NPS-AM-08-022

PROCEEDINGS

OF THE

FIFTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM WEDNESDAY SESSIONS VOLUME I

Acquisition Research: Creating Synergy for Informed Change

May 14-15, 2008

Published: 23 April 2008

Prepared for: Naval Postgraduate School, Monterey, California 93943

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
MAY 2008

2. REPORT TYPE

3. DATES COVERED
14-05-2008 to 15-05-2008

4. TITLE AND SUBTITLE
Proceedings of the Fifth Annual Acquisition Research Symposium Wednesday Sessions, Volume I. Acquisition Research: Creating Synergy for Informed Change Held at Montery, CA, May 14-15, 2008

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School,Graduate School of Business and Public Policy,555 Dyer Road, Room 332,Monterey,CA,93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**261**	

The research presented at the symposium was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret.)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.org

Conference Website:
www.researchsymposium.org

Preface and Acknowledgements

One hundred years ago, William H. Allen—of the then-newly founded Bureau of Municipal Research—issued a call in the *Journal of Accountancy* for “1,000 accountants for municipal research.” Reflecting the Progressive Era’s focus on domestic reform after the closing of the American frontier, Allen wrote:

> Only a pessimist will believe that the day is past for the pioneer. It is true that America has been discovered and that the law of diminishing returns long since began to operate in the gold fields of California, the wheat fields of the Northwest and the oil wells of Pennsylvania. It is also true that there is less opportunity today than ever before for adventure of the story book type. But to young men [sic] capable of thrilling with excitement when confronted with new problems to solve and new ideas to work out, I wish seriously to recommend a substitute for the North Pole—the unexploited field of municipal accounting and municipal business.[1]

Allen’s call followed the Bureau’s early and remarkable successes, both in exposing waste and corruption in New York City’s government and in devising and installing managerial systems for increased efficiency and transparency:

> [T]he mayor, comptroller, commissioner of street cleaning, president of Bellevue and allied hospitals, commissioner of parks and the commissioners of accounts have requested cooperation, and used departmental facilities and men [sic] for research and reorganization. We believe that similar cooperation will be obtained wherever private bodies or especially trained accountants approach the problem of municipal business with the sole motive of advancing the interests of the general public, and not with a desire to do sharpshooting, to turn up a scandal or to turn out the rascals.[2]

History has not recorded the extent to which Allen’s call for 1,000 accountants was answered. History has recorded, however, the very clear and significant contributions of his Bureau’s work—particularly its research agenda—to the formation of the field of American Public Administration.[3] Thus, while Allen certainly perceived the potential contributions of administrative research, it’s highly doubtful he could have imagined the development and maturation over the next century of this entirely new field of study in the US. Public Administration today includes hundreds of graduate degree programs, dozens of academic journals and conferences, and thousands of scholars. The objects of its study have expanded from municipal administration to include federal, state, international, and, more recently, not-for-profit administration.

[1] Allen, W.H. (1908, July). Wanted—One thousand efficient accountants for municipal research. *Journal of Accountancy, 6*, 187.

[2] Allen, 1908, pp. 192-193.

[3] Waldo, D. (1984). *The administrative state* (2nd ed.). New York: Holmes & Meier, p. 31.

Today, we issue a call for "1,000 scholars for defense acquisition research." As Allen believed in the possibilities for municipal research, we also believe in the possibilities for melioration of acquisition's seemingly intractable problems through systematic study. While Allen saw the skills of accountants as sufficient for the tasks he had in mind, we instead call for a truly interdisciplinary mix of scholars suitable for engaging the diverse facets of acquisition's many technical, managerial, and political issues.

Obviously, such an ambitious call cannot be answered by a single or even a dozen institutions. Accordingly, the NPS Acquisition Research Program has among its principal objectives the cultivation of an interdisciplinary community of acquisition scholars from many institutions around the world. This Symposium is merely a single step toward achieving that objective. Other recent steps include new research partnerships that NPS has forged with several other universities and the new *International Journal of Defense Acquisition Management* (http://www.acquisitionjournal.org), a scholarly journal jointly published and supported by the Acquisition Research Program and Cranfield University at the Defence College of Management of Technology.

From our limited perspective, such steps may seem woefully inadequate for the task of achieving meaningful and lasting acquisition reform. If so, we may do well to look forward into the next century and imagine our intellectual descendents who will study in a fully mature field of defense acquisition management and who will commend us for our efforts.

We gratefully acknowledge the ongoing support and leadership of our sponsors, whose foresight and vision have assured the continuing success of the Acquisition Research Program:

- Under Secretary of Defense (Acquisition, Technology and Logistics)
- Program Executive Officer (Ships)
- Program Executive Officer (Integrated Warfare Systems)
- Program Executive Officer (Littoral and Mine Warfare)
- Commander, Naval Sea Systems Command
- Deputy Assistant Secretary of the Navy (Acquisition and Logistics Management)
- Office of Naval Air Systems Command PMA-290
- Office of the Assistant Secretary of the Army for Acquisition, Logistics and Technology
- Director, Strategic Systems Program
- Project Manager Modular Brigade Enhancements
- Deputy Assistant Secretary Air Force (Management Policy & Program Integration)
- Dean of Research, Naval Postgraduate School

We also thank the Naval Postgraduate School Foundation and acknowledge its generous contributions in support of this Symposium.

James B. Greene, Jr.
Rear Admiral, US Navy (Ret.)

Keith F. Snider, PhD
Associate Professor

Karey L. Shaffer, MBA
Program Manager, Acquisition Research Program

The NPS A Team

Rear Admiral James B. Greene, Jr. USN (Ret.)—Acquisition Chair, Naval Postgraduate School. RADM Greene develops, implements and oversees the Acquisition Research Program in the Graduate School of Business and Public Policy. He interfaces with DoD, industry and government leaders in acquisition, coordinates graduate student projects and conducts guest lectures and seminars. Before serving at NPS, RADM Greene was an independent consultant focusing on Defense Industry business development strategy and execution (for both the public and private sectors), minimizing lifecycle costs through technology applications, alternative financing arrangements for capital-asset procurement, and "red-teaming" corporate proposals for major government procurements.

RADM Greene served as the Assistant Deputy Chief of Naval Operations (Logistics) in the Pentagon from 1991-1995. As Assistant Deputy, he provided oversight, direction and budget development for worldwide US Navy logistics operations. He facilitated depot maintenance, supply chain management, base/station management, environmental programs and logistic advice, and support to the Chief of Naval Operations. Some of his focuses during this time were leading Navy-wide efforts to digitize all technical data (and, therefore, reduce cycle-time) and to develop and implement strategy for procurement of eleven Sealift ships for the rapid deployment forces. He also served as the Senior Military Assistant to the Under Secretary of Defense (Acquisition) from 1987-1990; as such, he advised and counseled the Under Secretary in directing the DoD procurement process.

From 1984-1987, RADM Greene was the Project Manager for the AEGIS project. This was the DoD's largest acquisition project, with an annual budget in excess of $5 billion/year. The project provided oversight and management of research, development, design, production, fleet introduction and full lifecycle support of the entire fleet of AEGIS cruisers, destroyers, and weapons systems through more than 2500 industry contracts. From 1980-1984, RADM Greene served as Director, Committee Liaison, Office of Legislative Affairs followed by a tour as the Executive Assistant, to the Assistant Secretary of the Navy (Shipbuilding and Logistics). From 1964-1980, RADM Greene served as a Surface Warfare Officer in various duties, culminating in Command-at-Sea. His assignments included numerous wartime deployments to Vietnam, as well as the Indian Ocean and the Persian Gulf.

RADM Greene received a BS in Electrical Engineering from Brown University in 1964; he earned an MS in Electrical Engineering and an MS in Business Administration from the Naval Postgraduate School in 1973.

Keith F. Snider—Associate Professor of Public Administration and Management in the Graduate School of Business & Public Policy at the Naval Postgraduate School in Monterey, California, where he teaches courses related to defense acquisition management. He also serves as Principal Investigator for the NPS Acquisition Research Program and as Academic Associate for resident NPS acquisition curricula.

Professor Snider has a PhD in Public Administration and Public Affairs from Virginia Polytechnic Institute and State University, a Master of Science degree in Operations Research from the Naval Postgraduate School, and a Bachelor of Science degree from the United States Military Academy at West Point. He served as a field artillery officer in the US Army for twenty years, retiring at the rank of Lieutenant Colonel. He is a former member of the Army Acquisition Corps and a graduate of the Program Manager's Course at the Defense Systems Management College.

Professor Snider's recent publications appear in *American Review of Public Administration*, *Administration and Society*, *Administrative Theory & Praxis*, *Journal of Public Procurement*, *Acquisition Review Quarterly*, and *Project Management Journal.*

Karey L. Shaffer—Program Manager for General Dynamics Information Technology in support of the Acquisition Research Program at the Graduate School of Business and Public Policy, Naval Postgraduate School. As PM, Shaffer is responsible for operations and publications in conjunction with the Acquisition Chair and the Principal Investigator. She has also catalyzed, organized and managed the Acquisition Research Symposiums hosted by NPS.

Shaffer has also served as the Project Manager for Imagicast, Inc., and as the Operations Manager for the Montana World Trade Center. At Imagicast, she was asked to take over the project management of four failing pilots for Levi Strauss in the San Francisco office. Within four months, the pilots were released; the project lifecycle was shortened; and the production process was refined. In this latter capacity at the MWTC, Shaffer developed operating procedures, policies and processes in compliance with state and federal grant law. Concurrently, she managed $1.25 million in federal appropriations, developed budgeting systems and secured a $400,000 federal technology grant. As the Operations Manager, she also designed MWTC's Conference site, managed various marketing conferences, and taught student practicum programs and seminars.

Shaffer holds an MBA from San Francisco State University and earned her BA in Business Administration (focus on International Business, Marketing and Management) from the University of Montana.

A special thanks to our editors Jeri Larsen, Breanne Grover and Jessica Moon for all that they have done to make this publication a success, to David Wood, Tera Yoder and Ian White for production and graphic support and to the staff at the Graduate School of Business & Public Policy for their administrative support. Our program success is directly related to the combined efforts of many.

ACQUISITION RESEARCH: CREATING SYNERGY FOR INFORMED CHANGE

6th Annual Acquisition Research Symposium

May 13 - 14, 2009

Monterey, California

Announcement and Call for Proposals

The Graduate School of Business & Public Policy at the Naval Postgraduate School announces the 6th **Annual Acquisition Research Symposium** to be held **May 13-14, 2009 in Monterey, California**.

This symposium serves as a forum for the presentation of acquisition research and the exchange of ideas among scholars and practitioners of public-sector acquisition. We seek a diverse audience of influential attendees from academe, government, and industry who are well placed to shape and promote future research in acquisition.

The Symposium Program Committee solicits proposals for panels and/or papers from academicians, practitioners, students and others with interests in the study of acquisition. The following list of topics is provided to indicate the range of potential research areas of interest for this symposium: **acquisition and procurement policy, supply chain management, public budgeting and finance, cost management, project management, logistics management, engineering management, outsourcing, performance measurement, and organization studies.**

Proposals must be submitted by **November 7, 2008**. The Program Committee will make notifications of accepted proposals by **December 5, 2008**. Final papers must be submitted by **April 3, 2009** to be included in the Symposium Proceedings.

Proposals for papers should include an abstract along with identification, affiliation, and contact information for the author(s). Proposals for papers plan for a 20 minute presentation. Proposals for panels (plan for 90 minute duration) should include the same information as above as well as a description of the panel subject and format, along with participants' names, qualifications and the specific contributions each participant will make to the panel.

Submit paper and panel proposals to www.researchsymposium.org .

THIS PAGE INTENTIONALLY LEFT BLANK

NPS-AM-08-022

PROCEEDINGS

OF THE

FIFTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

WEDNESDAY SESSIONS

VOLUME I

Acquisition Research:

Creating Synergy for Informed Change

May 14-15, 2008

Published: 23 April 2008

Approved for public release, distribution unlimited.

Prepared for: Naval Postgraduate School, Monterey, California 93943

Disclaimer: The views represented in this report are those of the authors and do not reflect the official policy position of the Navy, the Department of Defense, or the Federal Government.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

Wednesday Sessions, May 14, 2008

Keynote Speaker

Wednesday, May 14, 2008 8:00 a.m. – 9:15 a.m.	**Keynote Speaker**
	Keynote Speaker **The Honorable Sue C. Payton**, Assistant Secretary of the Air Force for Acquisition

The Honorable Sue C. Payton is the Assistant Secretary of the Air Force for Acquisition, Washington, DC. She is the Air Force's service acquisition executive, responsible for all Air Force research, development and non-space acquisition activities. She provides direction, guidance and supervision of all matters pertaining to the formulation, review, approval and execution of acquisition plans, policies and programs. Payton directs $30 billion annual investments that include major programs like the KC-45A, F-22A, F-35, C-17 and munitions, as well as capability areas such as information technology and command and control, intelligence, surveillance and reconnaissance systems. She formulates and executes the $210 billion Air Force investment strategy to acquire systems and support services to provide combat capability to joint warfighting commanders.

Payton has previously worked with the military services, defense agencies, industry, coalition partners and combatant commands, and has had oversight responsibilities for technology transition programs. These programs include Advanced Concept Technology Demonstrations, Joint Warfighting Program, Foreign Comparative Test, Defense Acquisition Challenge, Technology Transition Initiative, ManTech, Defense Production Act Title III and TechLink. While working for ImageLinks, Inc., and the National Center for Applied Technology, she was responsible for the assessment, prototype development and insertion of commercial technology for Department of Defense agencies and worldwide field users. During her tenure with Lockheed Martin and Martin Marietta, Payton was responsible for leveraging the latest information systems technology needs of the DoD and the intelligence community, and she resolved complex acquisition and technical issues. Payton has extensive experience leading government and industry partnerships focused on maturing and applying technology, operations concepts, tactics, techniques and procedures to solve national security problems worldwide.

THIS PAGE INTENTIONALLY LEFT BLANK

Panel 1 - Plenary Panel - Maintaining Competition in Defense Acquisition

Wednesday, May 14, 2008	**Panel 1 - Plenary Panel - Maintaining Competition in Defense Acquisition**
9:30 a.m. – 11:00 a.m.	**Chair:** **Dr. Jacques S. Gansler**, Director, Center for Public Policy & Private Enterprise, University of Maryland, former Under Secretary of Defense for Acquisition, Technology & Logistics **Discussants:** **Dr. James I. Finley**, Deputy Under Secretary of Defense (Acquisition & Technology) **Louis A. Kratz**, Vice President and Managing Director, Focused Logistics, Lockheed Martin Corporation **Kenneth E. Miller**, Special Assistant for Acquisition Governance & Transparency to the Secretary of the Air Force

Chair: The Honorable Jacques S. Gansler is the Director of the Center for Public Policy and Private Enterprise and the Roger C. Lipitz Chair in Public Policy and Private Enterprise. From 1997 to 2001, he was the Under Secretary of Defense for Acquisition, Technology, and Logistics. As the third-ranking civilian at the Pentagon, Gansler was responsible for all research and development, acquisition reform, logistics, advanced technology, environmental security, defense industry, and numerous other security programs. Gansler has held a variety of positions in government and the private sector, including Deputy Assistant Secretary of Defense (Material Acquisition), Assistant Director of Defense Research and Engineering (Electronics), Senior Vice President, TASC, Vice President of ITT, and engineering and management positions with Singer and Raytheon Corporations. Gansler is a member of the National Academy of Engineering and a fellow of the National Academy of Public Administration. In addition to giving frequent congressional testimonies, he is the author of many articles and has written several books on the topic of defense.

Discussant: Dr. James I. Finley was confirmed as Deputy Under Secretary of Defense for Acquisition and Technology (A&T) by the US Senate in March 2006. In this role, he presides over policies that govern acquisition and procurement in the Department of Defense (DoD). He also advises the Secretary of Defense and the Under Secretary of Defense for Acquisition, Technology and Logistics on acquisition matters and technology integration and protection.

Finley came to the DoD in his current role after 30 years in the private sector, where he held various positions of increasing responsibility for General Electric, Singer, Lear Siegler, United Technologies and General Dynamics. He served in key positions on General Dynamics' leadership team, including Corporate Officer, President of Information Systems and Chair of the Business Development Council, before founding his own business-consulting company, The Finley Group, LLC, in 2002.

Finley's private-sector experience spans air, land, sea and space programs for the DoD, in addition to contributions to the Federal Aviation Administration's Automatic Surface Detection Radar System and the National Aeronautics and Space Administration's Space Shuttle Program. He also performed a variety of functions on systems and subsystems, including: mission analysis; design, development and deployment of weapon delivery; flight control; navigation; information management; command, control, communications, computer, intelligence, surveillance and reconnaissance (C4ISR); battle space management; and chemical/biological defense systems. Additionally, Finley has extensive private-sector experience working on Joint programs—for example, the Joint Tactical Information Distribution System and the Joint Surveillance and Target Attack Radar System.

Finley holds a Bachelor of Science in Electrical Engineering from the Milwaukee School of Engineering, and he earned a Master’s of Business Administration from California State University, Fresno. The Milwaukee School of Engineering also awarded him an Honorary Doctorate in Engineering.

In addition to his academic achievements, Finley is the recipient of numerous professional performance awards, including the Boeing Gold Certification Award, the Honeywell Preferred Supplier Award, Northrop Grumman Blue Achievement, Lockheed Martin Best in Class Rating, the Defense Security Service "Outstanding Achievement" Award, the George Westinghouse Award and the California State University "2007 Top Dog Alumni Award."

Finley resides in Virginia with his wife, Sharon. They have six children and four grandchildren.

Discussant: Louis A. Kratz is the Vice President and Managing Director, Focused Logistics, Lockheed Martin Corporation, responsible for coordinating Lockheed Martin's logistics and weapon system sustainment efforts. Kratz leads Lockheed Martin's logistics strategic planning, performance-based logistics efforts, logistics technology development, logistics human capital development, and cross-corporate logistics business initiatives.

Previously, Kratz served as the Assistant Deputy Under Secretary of Defense (Logistics Plans and Programs) within the Office of the Deputy Under Secretary of Defense (Logistics and Materiel Readiness). As such, he was responsible for guiding the DoD's logistics transformation to meet the operational requirements of the 21st Century. Kratz oversaw the DoD's long-range logistics planning to meet the requirements of the *Quadrennial Defense Review* (*QDR*) and *Joint Vision 2020*. He led the core analytic team on supply chain logistics for the *QDR* and prepared the DoD's inaugural Focused Logistics Roadmap. Kratz led the DoD's implementation of Total Life Cycle Systems Management and Performance-based Logistics, including acquisition logistics policy development, career development, and oversight of major weapon systems. Kratz was the Defense Standardization Executive and co-chair of the Focused Logistics Functional Capabilities Board and the Joint Logistics Group.

Prior to that, Kratz was the Director of Life Cycle Integration at TASC. Focused on weapon systems acquisition, acquisition reform and information resource management. Kratz led TASC's horizontal integration effort on weapon system support and directed TASC's support to the OSD Acquisition Reform Office and the FAA Acquisition Policy Office, including policy development, metrics, cost/benefit analyses, and best practices assessments. Kratz directed detailed acquisition strategy analyses for numerous weapon system programs.

Kratz is a member of several organizations, including the National Defense Industries Association (NDIA) and Aerospace Industries Association (AIA), and serves as chair of the Logistics Education Board of Advisors for SOLE—The International Society of Logistics. He was the inaugural recipient of the 2005 Von Braun Award for Leadership from SOLE, and has received the 2004 Presidential Rank Award and the AIA Outstanding Award.

Discussant: Kenneth E. Miller, a member of the Senior Executive Service, is Special Assistant for Acquisition Governance and Transparency to the Secretary of the Air Force, Washington, DC. Miller assists in discharging the responsibilities in the direction, guidance and supervision of Air Force programs for research, development and acquisition of systems, supplies and services. This includes the formulation of acquisition and contracting policies and the management oversight of specific acquisition programs.

Miller, a native of Columbus, MS, began his professional career in 1975 as an aerospace engineer with the Naval Air Systems Command. He advanced to weapons systems acquisition management as the Assistant Deputy Program Manager for the H-3 antisubmarine helicopter, later serving as Deputy Program Manager for the E-6A and Principal Deputy Program Manager for the A-6/EA-6 Weapons Systems Program Office. In April 1989, the Navy established the new Program Executive Offices within the acquisition system. Miller was selected to be the first Deputy for Acquisition for the Program Executive Office (Tactical Aircraft), providing policy and execution advice to the Program Executive Officer on assigned programs.

Miller was appointed to the Senior Executive Service as the second Deputy Program Executive Officer for Tactical Aircraft Programs, providing advice on acquisition-related issues for a variety of aircraft and weapons programs. In 1994, he was selected as the Assistant Commander for Corporate Operations, where his responsibilities included the strategic planning and corporate business functions of the Naval Air Systems Command. Additional duties included Chief Information Officer. In 1998, he was appointed Principal Assistant for Acquisition, Programming and Budgeting for the Director of Air Warfare within the Office of the Chief of Naval Operations. Miller was later selected as the Assistant Deputy, Chief of Naval Operations, Warfare Requirements and Programs, defining and developing a variety of warfare requirements for the Department of the Navy. He is a frequent speaker at government, industry and national forums.

THIS PAGE INTENTIONALLY LEFT BLANK

Panel 2 - Acquisition Portfolio Management

Wednesday, May 14, 2008

11:15 a.m. – 12:45 p.m.

Panel 2 - Acquisition Portfolio Management

Chair:

Dr. Nancy Spruill, Director, Acquisition Resources & Analysis, Office of the Under Secretary of Defense (Acquisition Reform)

Papers:

GAO Best Practices: Portfolio Management

Mr. Michael P. Sullivan, Director, Dayton Ohio Office, US Government Accountability Office

Portfolio Management Best Practices for the DoD, R&D and Acquisition Communities

Mr. Robert Edson, Principal Analyst and Technical Manager, Innovation Analysis Division and **Nicole Long**, Associate Analyst, ANSER

GAP Analysis: Rethinking the Conceptual Foundations

Mr. Gary O. Langford, Lecturer, **Dr. Thomas V. Huynh**, Associate Professor, **Dr. Raymond (Chip) Franck**, Senior Lecturer and **Dr. Ira A. Lewis**, Associate Professor, Naval Postgraduate School

Chair: Dr. Nancy Spruill is a native of Takoma Park, MD. After receiving a Bachelor of Science Degree in Mathematics from the University of Maryland in 1971, she joined the Center for Naval Analyses (CNA). From 1971 to 1983, she held a variety of positions on the CNA staff, including Technical Staff Analyst, Professional Staff Analyst and Project Director. In 1975, she earned her Master of Arts in Mathematical Statistics from George Washington University, followed by her Doctorate in 1980.

Spruill served on the staff of the Office of the Secretary of Defense from 1983 to 1993. Initially, she was the Senior Planning, Programming, and Budget Analyst in the Manpower, Reserve Affairs and Logistics Secretariat. Later, she served as the Director for Support and Liaison for the Assistant Secretary of Defense for Force Management and Personnel. Then, she served as the Senior Operations Research Analyst in the Office of the Assistant Secretary of Defense for Program Analysis and Evaluation.

In 1993, she joined the staff of the Defense Mapping Agency (DMA), serving as the Chief of Programs and Analysis Division for the DMA Comptroller. Subsequently, she served as Acting Deputy Comptroller and was a member of the Reinvention Task Force for the Vice President's National Performance Review.

In March 1995, she was selected as the Deputy Director for Acquisition Resources for the Under Secretary of Defense for Acquisition and Technology. In February 1999, she was appointed Director, Acquisition Resources & Analysis (ARA) for Under Secretary of Defense for Acquisition, Technology and Logistics (USD(AT&L)). In this capacity, she is responsible for all aspects of AT&L's participation in the Planning, Programming and Budgeting System (PPBS); the Congressional process; and the Defense Acquisition System. She serves as the Executive Secretary to the Defense Acquisition Board and is responsible for the timely and accurate submission to Congress of Selected Acquisition Reports and Unit Cost Reports for Major Defense Acquisition Programs. She manages the Defense Acquisition Execution Summary monthly review of programs; monitors cost and schedule status of high interest programs; and conducts analyses of contract and program cost performance, including analysis of the effective use of Integrated Program Management principles through the use of Earned Value Management. Spruill performs systemic analysis to improve acquisition policy and education, and conducts special analyses for the Under Secretary. She leads the Department in developing plans to manage Property, Plant and Equipment, Inventory, Operating Materials and Supplies/Deferred Maintenance and Environmental Liabilities. She proposes modifications to (or acquisition of) new DoD feeder systems, in support of achieving an unqualified audit opinion on DoD Financial Statements as mandated by the *Chief Financial Officers* (*CFO*) *Act*. She also manages the studies program for OSD, oversees USD(AT&L)'s office automation system, manages its information system network, and conducts special analyses for the Under Secretary.

Spruill has been a member of the Senior Executive Service since 1995. She is a certified Acquisition Professional and an active member of the American Statistical Association. Her many honors and awards include the Department of Defense Medal for Distinguished Civilian Service, the Secretary of Defense Medal for Exceptional Civilian Service, the Secretary of Defense Medal for Meritorious Civilian Service, the Hammer Award, the Acker Skill in Communications Award and the Presidential Rank Award. She has contributed papers in publications of the statistics and defense analyses communities and authored articles in the general press on how politicians use—and abuse—statistics.

Gap Analysis: Rethinking the Conceptual Foundations

Presenter: Gary Langford is a Lecturer in the Systems Engineering Department at the Naval Postgraduate School in Monterey, California. He teaches systems engineering, combat systems, and project management. His research interests include the theory of systems engineering and its application to military and commercial competitiveness. Additionally, Langford founded and ran five corporations—one NASDAQ listed. He was a NASA Ames Fellow. He received his MS in Physics from Cal State, Hayward, and his AB in Astronomy from the University of California, Berkeley.

Gary O. Langford
Department of Systems Engineering
Graduate School of Engineering and Applied Sciences
Naval Postgraduate School
222 Bullard Hall
777 Dyer Road
Monterey, CA 93943-5000
Email: golangfo@nps.edu

Author: Raymond E. Franck is a Senior Lecturer in the School of Business and Public Policy, also at the Naval Postgraduate School (NPS). He teaches economics (generally) and pursues research primarily in the areas of military innovation and defense acquisition. Prior to joining the NPS faculty, Franck served for 33 years in the Air Force, retiring as a Brigadier General in 2000. His NPS service has included Interim Chair, Systems Engineering Department from 2002 to 2004. He received his undergraduate degree from the Air Force Academy and his PhD from Harvard University.

Raymond E. Franck
Graduate School of Business and Public Policy
Naval Postgraduate School
203 Ingersoll Hall
555 Dyer Road
Monterey, CA 93943-5103
Email: refranck@nps.edu

Author: Thomas Huynh is an Associate Professor in the Department of Systems Engineering at the School of Engineering and Applied Sciences, Naval Postgraduate School (NPS). He teaches systems engineering, project management, and acquisition. His research interests are uncertainty management in systems engineering, system scaling, and simulation-based acquisition. Prior to joining the NPS faculty, Huynh worked for 23 years as Fellow, Modeling & Simulation and Information Sciences, Advanced Technology Center, Lockheed Martin Space Systems, Palo Alto. He is Academic Associate for the NPS Joint Executive Systems Engineering Management Program. He received his BS degrees in Chemical Engineering and Mathematics from University of California, Berkeley, and his MS and PhD in Physics from UCLA.

Thomas V. Huynh
Department of Systems Engineering
Graduate School of Engineering and Applied Sciences
Naval Postgraduate School
Bullard Hall
777 Dyer Road
Monterey, CA 93943
Email: thuynh@nps.edu

Author: Ira A. Lewis is an Associate Professor of Logistics in the School of Business and Public Policy at the Naval Postgraduate School (NPS). He teaches logistics and operations management and pursues research primarily in the areas of military logistics and financial operations. Prior to

joining the NPS faculty, Lewis was Research Officer for Supply Chain Project in the Department of National Defence, Ottawa, Canada. His NPS service has included personnel management and academic administration in the School, and Associate Dean for Faculty Affairs (2002 to 2006). He received his MBA from University of Ottawa and his PhD from Arizona State University.

Ira A. Lewis
Graduate School of Business and Public Policy
Naval Postgraduate School
Ingersoll Hall
555 Dyer Road
Monterey, CA 93943-5103
Email: ialewis@nps.edu

Abstract

Gap Analysis is widely regarded as a useful tool to facilitate commercial and defense system acquisitions. This paper is a rethinking of the theoretical foundations and systematics of Gap Analysis with practical extensions to illustrate its utility and limitations. It also provides a new perspective on those theoretical foundations from the perspectives of systems and value engineering.

The growing sophistication and complexity of new systems or system-of-systems have resulted in a dramatic increase in time and money to reach operational capability. Gap Analysis, properly defined and enacted, clarifies goals, appropriate investment and the end-use.

Introduction

The challenge of successfully acquiring and operating a new system is to ensure that the mission will be accomplished within an acceptable level of loss. To that end, there have been numerous attempts to develop and field systems that are intended to prevail in the event of conflict. How should these future systems be defined? Who is responsible? What processes guide the system requirements? If “we” perceive a deficiency or a desired goal that is different from that which we are intending, then there could exist a basis for gap in capability and, therefore, a desire to close the capability gap.

What one desires versus what one has is, in essence, a Gap. The Gap is as much the relationship between what is perceived to be *important* and the derived difference between performance and expectations. The methodology and analysis of that difference is the descriptive foundation for Gap Analysis. From a mission-capability perspective, a Gap may consist of deficiencies in operational concepts, current or projected operational disadvantages, technologies, and understood future needs. To be specific, a Gap must be founded on the starting and ending points as well as the difference between these points. Quantifying these metrics typically involves evaluating a number of situations and mission scenarios in concert with actions or, more generally stated, guidance from policy and goals. Measures of Effectiveness (MOEs) have long formed the set of standards from which to determine how well a capability satisfies a requirement (Sproles, 2002). MOEs are distinguishable from Measures of Performance (MOPs) in that MOEs offer the external view, while MOPs are more consistent with the internal view. The external view captures the system’s beginning and ending points, and the MOE of the candidate programs to fill the gap. The internal view involves measures of how well one fills the gap through the MOPs.

Therefore, one must formulate both MOEs and MOPs to fully define a Gap. However, Chairman of the Joint Chiefs of Staff Instruction *CJCSI 3170.01D*, *Joint Capabilities Integration and Development System* (2004, March 12) focuses on MOEs, yet does not mention MOPs. There is an implied admixture of MOEs and MOPs defined as MOEs, but the essential qualities of performance-based metrics are missing for carrying out activities and actions, for measuring functions, and from which to determine economic and numeric losses.

Gap Analysis is deeply embedded and fully institutionalized as a cornerstone of the United States Department of Defense (DoD) acquisition strategy, particularly in the critical process called Valuation of Alternative (VoA), formerly Analysis of Alternatives (AoA). It is the purpose of Gap Analysis within the DoD VoA process to report on the evaluation of the performance, operational effectiveness, operational suitability, and estimated costs of the alternative systems to meet a desired mission capability. In this context, the VoA assesses the advantages and disadvantages of alternatives under consideration.

The goal of Department of Defense (DoD) Gap Analysis is to compare current capability to a set of requirements. Where differences arise, gaps are identified and quantified, and mitigations are prioritized and planned. This paper addresses the theoretical foundations and systematics of Gap Analysis with extensions to illustrate its utility as a useful management tool for both defense and commercial acquisition purposes. Without a considered theoretical foundation from which to conduct Gap Analysis, an inadequate level of guidance regarding appropriate methodology and analytical methods may well result. The metrics of Gap Analysis are defined on the basis of system value (Langford, 2006; Langford & Huynh, 2007) and assessed risks.

Discussion

For the US Department of Defense (DoD), the acquisition of goods and services follows the policies outlined in Directives, the Joint Capabilities Integration and Development System (JCIDS), and the Defense Acquisition Guidebook *DoD 5000* (the structure and operation of the defense acquisition system). In this context, Gap Analysis is a method for identifying the degree to which a current system satisfies a set of requirements. The goal of Gap Analysis is to align an anticipated future outcome with a future reality that can be formulated, definitized, and established or constructed. However, Gap Analysis is not intended to close the space between the most distant extremes or the rarest occurrences. Rather, Gap Analysis is centered on the larger, more general aspects that are by and large not part of the present reality (referred to as the current reference frame). For the DoD, Gap Analysis grew out of the realization that relying solely on a threat-based approach (used as a primary driver of requirements until 2000) or a technology approach to determining future needs is both costly and largely ineffective. One of the concerns with threat-based methods lies with the notion of being guided by the will and intentions of others (e.g., adversaries) (as exemplified through an analysis of threats, their efficacy and robustness), rather than relying on our competitive advantages to define and frame future engagements.

Alternatively, a technology-centered approach is open-ended, with little constraint for what can be postulated. Acquisitions based on a technology approach may not result in a continuous presentation of appropriate military hardware that is consistent with lifecycle cost issues or the necessary capabilities in time of conflict. With only theoretical physics as the constraint, technology developments can extend twenty years or longer (e.g., ground-based, airborne, and space-based laser weapon systems). Even with an incremental approach to delivering products, usable incarnations of systems may be distanced by inadequacies in the phases of development and levels of integration. At issue is the availability of weapons systems and doctrine that can prevail in hostilities without: (1) spending an enormous amount of money to sustain existing systems until new systems are delivered, and (2) having to develop a *needed* technology engineered and made available for use. Acquisitions based on a threat-based approach are always plagued by the credibility of the threat—the absolute measure of what an adversary will have available in the future.

Accordingly, neither the technology nor the threat-based approaches address some of the persistent, perennial issues that fundamentally impact the implementations of Gap Analysis.

Since the turn of the century, the DoD has concentrated on a capabilities-based approach, in which capabilities are defined and identified using a top-down approach infused with characteristics of measures of effectiveness, supportability, time, distance, effect (including scale) and obstacles to overcome. Capability is defined by an operational user as the ability to execute a specified course of action (CJCS, 2004).

Gap Analysis Background

The first reference to Gap Analysis was in the 1938 publication on the disjuncture between cultural goals and institutional norms (Merton, 1938). The notion was adapted to psychotic behavior (1950s), preferred biodiversity (1980s), personnel planning (1989), and more recently, competitive analysis and interest rates of financial instruments. Gap Analysis was referred to in a series of instructions from the Chairman of the Joint Chiefs of Staff throughout the 1990s with reference to defining gaps in capabilities requiring a material solution. In the late 1990s, the DoD infused a form of Gap Analysis into the acquisition process—comparing future threat-based assessments to current capability. Meanwhile, program costs seemed out of control; major projects were cancelled, and functionality was not being delivered as desired. A memorandum from Secretary of Defense Donald Rumsfeld asked for ideas to fix the DoD process of determining system requirements (Rumsfeld, 2001). Gap Analysis had come into being and was thriving within the structure of the JCIDS process. The determinant factor for acquisition had moved from a threat-based premise to a capabilities-based identification of needs. While the threat continues to be a part of the acquisition process, part of the initial capabilities document (ICD)—the document that initiates the acquisition system management process—Gap Analysis is performed on the basis of desired capability.

While Gap Analysis should be neither technology-driven nor threat-driven, it is an approach that largely uses technology and threats as inputs to a vision-driven future. Gap Analysis is based on the high-level collective vision of *what we need*. This vision is discussed at the top levels of government within the context of the national security strategy: the strategic concepts postured for defense, the joint operations concepts, and the integrated architectures of US forces. The vision is then stated as a goal, one that is to be achieved methodically through a step-wise process. The problems with the existing formulation of Gap Analysis are determining: (1) what constitutes the foundation data, (2) which data are relevant to a future competitive analysis, (3) how should the relevant data be structured to deal with the future issues within the proper context, (4) what are the assumptions and scaling rules used to extend the current state of industrial output, technology advances, and engineering developments, and (5) what process or methodology enforces consistency of performing Gap Analysis.

Research Objectives

The process of identifying needs and unsatisfied desires, or gaps in capability—in essence, the goal—is sometimes enacted through a set of ad-hoc processes and actions accompanied by an analysis of alternatives. Closing the capability gap between what exists and what is wanted includes the aptitude required to develop something new and the reference point from which one starts. This is an Omni-dimensional problem that encompasses strategy, operations, systems engineering processes, and the compositional elements of the system. Technology readiness and maturity are integral parts of Gap Analysis.

The first step in improving Gap Analysis is to determine the underlying premises and fundamental metrics of such an Analysis. This paper investigates the theoretical issues of Gap Analysis and proposes seven metrics based on quantifiable worth, value, and risk. By developing the theory of Gap Analysis into a form that can be applied in a clear and consistent manner to the DoD acquisition process, we can directly apply the process of defining requirements. Specifically, Worth metrics can be applied to a critical examination of foundation data; Risk metrics can be used to interpret the relevancy of data; an Enterprise Framework (which displays Worth and Risk metrics) can illustrate context at a given time; and assumptions can be definitively scrutinized. To further understand and determine the applicability of Gap Analysis for DoD acquisition, a final step in this work is to identify the general limitations of Gap Analysis and the general impositions that Gap Analysis places on the success of the acquisition process.

Theory

Gaps have to do with mechanical causal histories—the telelogic argument that gaps exist and can be ameliorated by goal-directed actions. Aristotle was the first philosopher to formulate an accountable theory of telelogy founded on four causal properties: material, formal, efficient, and final. He argued that these four causes are required to give a complete account of any event. The cause of material involves being made of matter (e.g., the product); the cause of formal involves relations between entities (e.g., the network); the cause of efficient involves acting in certain ways (e.g., the procedures); and the cause of final involves having specific goals towards which actions are directed (e.g., the use) (Aristotle, 350 B.C.E.).

For the Department of Defense, Gaps are defined in terms of functional areas, relevant span and domain of military operations, intended effects, temporal matters, policy implications and constraints. Further, all gaps are defined in terms of capability. The Joint Capabilities Integration Development System (JCIDS—the formal US DoD procedure which defines acquisition requirements and develops the criteria to evaluate weapon systems) was implemented to specifically address capability gaps. But not all capability gaps require a material solution set. Changes or enactments of Doctrine, Organization, Training, Materiel, Leadership and education, Personnel, and Facilities (DOTMLPF) are also considered to close Gaps. Such considerations are formally evaluated before recommending the start of a new acquisition effort (CJCS, 2004; CJCS, 2005). In essence, functional capabilities are assessed to identify gaps.

It is relevant to mention the pioneering work by Lawrence Miles (1972) to formally recognize and focus attention on functions of a product. Product functions create (or cause) performances relative to investment. The ratio of performance to investment is a measure of relevancy and effectiveness.

Yet Gap Analysis is concerned with the difference between the present and the future, the reality and the expected, but not with the time or discrete time-steps between these disparities. While the DoD typically formulates its development interests in a temporal domain (e.g., a timeline of activities), the development activities are construed and managed as a discrete set of events. The Systems Engineering Process Models reinforce the notion of when to move from one stage of product development to the next stage, as well as what tasks need to be completed within each stage. Consequently, the notion of a temporal juxtaposition of activities is less relevant to the event-driven outcomes that characterize a future competitive space. In other words, Gap Analysis does not reflect when something will actually happen, only that it will happen. This defining of a Gap (in a different way than found in US acquisition policy, *DoD 5000 series)* lends itself naturally to a display of intentions that accurately reflect the constraints of event-based competition founded on the product, operations, and strategy of the various competitors.

In total, this redacting of Gap Analysis into events rather than timelines eliminates the actual propositional attributes of the competition, but retains the notional attributes. Propositional attributes iterate the validity of belief attitudes (i.e., I know what I know; I know what I want). Notional attributes include intentions and wishes (i.e., the end result is not influenced by the proposer's illative skills) (Duzi, 2002). Temporal considerations (e.g., I know when I want it) can be added as an attribute of the Enterprise Framework after the researcher gains an understanding of the situational awareness in event-space (a structure and analysis formed without temporal constraints). There are alternative interpretations of Enterprise Frameworks, most notably for software applications (Hafedh, Fayad, Brugali, Hamu & Dori, 2002). This study maintains that such theories can be used to surmise a means to enforce consistency in process, application, and interpretation of Gaps.

Value

The prime distinguishing characteristic of Value Engineering is the use of functional (or function) analysis (Miles, 1972, first published 1961). Value Engineering (VE) was developed by Miles and Erlicher at General Electric in 1947 as a means recognize and explore what an element of the system does rather than what it is. Value Engineering is an organized process to optimize a system's functionality versus cost. Alternatively, VE provides the necessary functions at the lowest cost, or determines which alternative design will provide the most reliable performance for a given cost. In essence, analyzing Value is the way and manner of analyzing productivity, selecting alternatives, and otherwise manipulating the ratio of Performance to Cost. For the purpose of this report, the authors will not distinguish between Value Engineering and Value Analysis. Value Analysis (VA) is typically concerned with productivity, the use of labor, materials and profitability.

The term Value has many colloquial definitions, including the term's use and misuse often disguised as promoting various marketing and sales concepts. But in the main, constructs of Value are without merit and meaning unless there is a relationship to functions, and therefore by reference, to system objective(s) or use(s). Value is not synonymous with cost or investment. Value is the functionality and performance of a system divided by the investment to deliver or sustain that system (Langford, 2006; 2007). Further, Value is not Worth, which is a measure of Value given risk (discussed in next section). There are different types of Value (use, esteem, cost, exchange, scrap, and so forth). For the purposes of this report, the authors do distinguish between the types of Value.

Value compares the functionality and use one receives versus what one invested (Langford, 2006). This notion of Value explicitly requires a buyer and seller model to determine Value. This presupposes that there is always a “source and a sink,” an “input and an output,” a pre-condition and a post-condition that is the determinant of Value. Therefore, Value is the ratio of the defining characteristics of the product (Functions and Performances) divided by the investment to achieve that functionality and performance. Value is measured in absolute terms. For example, the product shall provide a function with a specified performance. That function does 0.5 of what was paid for (as perceived from the point of view of the developer). Or perhaps, the performance was measured at 90% of the requirement. The investment expended to achieve that functionality and performance was as planned. Therefore, the value was less than desired (developer’s perspective). The *Value Function* (Equation 1) relates the System Value to the System Use(s) or to the System function(s) and performances related to the functions, divided by the investment.

Equation 1.

$$V(t) = \frac{\sum\{F(t) * P(t)\}}{I(t)}$$

where $F(t)$ is a function performed by the system; $P(t)$ is the performance measure of the function $F(t)$; $I(t)$ is the investment (e.g., dollars or other equivalent convenience of at-risk assets); and the time, t, is measured relative to the onset of initial investment in the project. The unit of *V(t)* is that of $P(t)$ divided by Investment (which could be quantified in terms of dollars or another meaningful measure an investment), since $F(t)$ is dimensionless.

The importance of functions was underscored in 1954 by Lawrence Miles when he conceived Value Analysis (and the subsequent development of the fields of Value Engineering and Systems Engineering) away from the parochial focus of simply providing system components. He based his functional analysis on the component parts of the product, the totality of which provided the desired functions. The purpose of functional analysis was to establish why an element exists so that alternative solutions could be generated (Green, 1996). Value Engineering is the activity which identifies and analyzes the function of products and services to achieve an overall effectiveness in providing system functionality. Systems Engineering is the activity which identifies and analyzes functions of products and services to specify the requirements that need to be built and sustained.

When applied to Gap Analysis, the metrics used for analyzing requirements are Value and Risk. Value is captured by the cost of Functions and their Performances, and Investment (measured in cost or investment). In common-sense fashion, Value is a measure of appreciation. It may be objective or subjective. Objective value relates to the idea that there is independence of assessments viewed from various perspectives—a consensus opinion of truth. Subjective value is based on what is expected (the sum of all corporal and abstract happenings from which you benefit and expect from a situation if you participate in a certain fashion). Value is simply the matter of minimizing cost or its time equivalency to develop a product. Value is the use that users expect (e.g., the functions and performance) for the investment they are willing to make. Further, Value is exemplified in the formulation of lifecycle costs and lifecycle time that express the transformation of company assets into profitably sold products that have a set of functions, performances, and quality. Each function is an activity that the product does with certain performance attributes. For each function, there can be several performance requirements. But there is never a function without a performance (Miles, 1954; Langford, 2006; 2007).

Worth

The notion of Worth extends the concept of value to include the intangibles of an indefinite quantity or other uncertainties. We define Worth as an indefinite quantification of something having a specified value. For example, "I have $20. Please give me $20 worth of gasoline." I have already determined that gasoline has sufficient value to warrant my purchasing it, and I have a limited budget of $20. I am willing to purchase more gasoline at a later time, when I either have more than $20, have more time to pump the gasoline, or have a current additional constraint removed. But, is it worth it to purchase from this vendor or another vendor at another location? Perhaps $20 purchases more gasoline at a different location from a different vendor. The $20 will purchase either Quantity X from Vendor A or Quantity Y from Vendor B, where Quantity Y > Quantity X. The difference in distance between Vendor A and Vendor B is 5 miles, so I must drive an additional 5 miles to transact and receive more gas than I could receive from Vendor A. This presumes I know with a high degree of certainty that Vendor B offers the gasoline appropriate for my use and provides similar performance at a price sufficiently lower than I can get from Vendor A so as to warrant travel to Vendor B. If my level of knowledge was lower about the price of Vendor B, then I must consider the worth issue in light of the uncertainty that Vendor B's price would be sufficiently less than Vendor A's price. In other words, is it worth the risk to drive farther and "shop" for gasoline? The loss may be quantifiable in the case in which Vendor B's price is known. Either the price is sufficiently lower to justify driving to Vendor B's location or it is not. If both the price and the distance are unknown, then there is less sufficiency and greater unknowns with which to deal. These unknowns can be incorporated into the Worth function as a determination of losses. If I do not locate a gasoline vendor before I "run out of gas," I will incur additional costs of purchasing a gas can and the cost of my time converted on a cash basis. Further, if I locate a gasoline vendor whose price is higher than Vendor A, then I have paid more than I could have paid.

By including the effects of high, sufficient, and low measures of quality, a decision based on Worth can be structured and evaluated in a methodical fashion. Obtaining sufficient information is typified by the trade-off between when one has paid too much or too little for either a given number of defects (1) as measured by a degradation or improvement in performance, or (2) which result in defects that are caused by certain levels of performance.

Worth as simply the ratio of the Value *V(t)* multiplied by the Quality *Q(t)* (Langford, 2006; 2007). Performance indicates how well a function is executed by the system. In this work, quality refers to the consistency of performance (or tolerance that signifies how good the performance is) in reference to the amount of pain or loss that results from the inconsistency as described by Taguchi (1990). In essence, functions result in capabilities; performances differentiate competing products; and quality affects the lifecycle cost of the product. For each function, there is at least one pair of requirements—performance and quality. The quality requirement indicates the variation and impact of the variation of the performance requirement of a function. A system function may thus have different values of performance, and the quality of a performance may have different values. The summation in Equation 1 is, thus, over all values of the functions, performance, and quality, for all time, and incorporating all uncertainties. Equation 2 indicates the Worth of system, as it references Value.

Equation 2.

$$W(t) = V(t) * Q(t) = \frac{\sum F(t) * P(t) * Q(t)}{I(t)}$$

where $Q(t)$ is quality (the tolerance assigned to the performance measures), and the time, t, is measured relative to the onset of initial investment in the project. We refer to the delineation of a function in terms of its performance and the quality of the performance as the triadic decomposition of the function. If the unit of $Q(t)$ can be converted to the unit of $I(t)$ (Equation 1), then the unit of W(t) is that of $P(t)$, since $F(t)$ is dimensionless. $Q(t)$ can be thought of as a loss that is incurred.

Several schemes have been proposed to define and structure requirements, such as functions, performance, and tolerances/physical synthesis by Wymore (1993), hierarchical task analysis by Kruchten (2000), decomposition coordination method of multidisciplinary design optimization by Jianjiang, Zhong, Xiao, and Sun (2005), functional descriptions by Browning, Deyst, Eppinger and Whitney (2003) and Cantor (2003), and non-functional descriptions by Poort and Peter (2004). The functional triadic decomposition proposed in this work forms a basis for a management tool that provides a structure to control the project. Again, triadic decomposition prescribes that every function is imbued with the necessary and sufficient attributes of performance and quality. It forms a basis for a management tool that provides a structure to control the project.

Control centers on three functions (again, each with associated performance and quality): Regulate (monitor and adjust), govern (define limits, allocate resources, determine requirements, and report), and direct (lead, organize, and communicate).

Traditional functional analysis, supplemented with the triadic decomposition, is conjectured to result in a complete and comprehensive set of requirements. The resulting functional decomposition, together with commensurate system specifications and the mechanisms of action or activity (e.g., creation, destruction, modulation, translation, transduction), should form a basis upon which a system can be designed and built using the classical set of system development models—such as the spiral, “Vee,” and waterfall model.

The Value of a product is thus quantified according to Equation 1, and the Worth of a product is quantified according to Equation 2. From the manufacturer’s point-of-view, a “product’s worth” is one that has met some investment criteria for the desired set of functionality, performance, and quality requirements. From the purchaser’s (consumer’s) point-of-view, the expression in Equation 2 aids in the trade between the applicability of a purchased product (in terms of the item’s functionality, performance, and quality) and the total cost and time invested in the purchase and use of the product (Langford, 2006; 2007).

Value and Worth are calculated at the moment of the agreed exchange of product/services for a given amount or recompense. Worth reflects the uncertainties based on losses that are associated with the exchange. These exchanges (or interactions between elements) are quantifiable and may have a net impact on the Value and Worth of the system, or in the exchange between two or more systems through their respective elements, a system(s)-of-systems interaction. We are interested in the interactions that have consequences that are measurable in the lifecycle of the product or service. To incorporate this level of minimum interest, we introduce the concept of a Net Impact—defined as a consequence that exceeds a threshold that is determined to be of interest (Langford, 2006; 2007).

Worth Transfer Function

In control theory, a transfer function is a mathematical representation of the relation between the input and output of a system. A Worth transfer function (WTF) between two elements of a system is defined to be the exchange of Worth between the two elements. Worth is what is received (in terms of usefulness) for an investment. This exchange necessarily assumes some measure of risk. Given risk, a WTF can thus be either a manifestation of the state (or a change in the state of a system), or a tool to evaluate differences between the state of a system and the state of another system, or between the states of two systems in a system-of-systems. In essence, the WTF represents various impact(s) on the state(s) of a system. The WTF can be a nested hierarchy of WTFs, all related through functional decomposition. Depending on the worth ascribed to each of the WTFs, the state(s) of the system(s) may be impacted to varying degrees. The result is that a small number of WTFs may be equivalent to a large number of irreducible WTFs.

A system is a set of elements that are either dependent or independent but interacting pairwise—temporally or physically—to achieve a purpose. The elements form the boundary of the system. This definition takes into account both the permanent and episodic interactions among elements of a system or systems of a system-of-systems. It thus includes the lasting and occasional interactions, as well as emergent properties and behaviors, of a system. These interactions effect transfer of energy, materiel, data, information, and services. They can be cooperative or competitive in nature, and they can enhance or degrade the system Worth, which is defined below. The pairwise interaction transfers a measure of Worth from one element of a pair to the other element. We term the measure of the transferred worth the Worth Transfer Function (WTF).

Complexity

Complexity of a system is often characterized by the total quantity of units that make up the system. As described by Homer and Selman (2001) and Li and Vitanyi (2006), it is both the number of and interactions among the units that, in general, are used to imply and define complexity. The system complexity thus augments the management challenge because of the large number and various types of system elements and stakeholders. In this work, complexity is reflected by the number and significance of WTFs among the elements of a system or among the systems of a system-of-systems. Since an element of a system may also be a stakeholder of the system, an increase in the number of stakeholders increases the complexity. Managing complexity or managing stakeholders thus amounts to managing the WTFs. It must be noted that a stakeholder with a large WTF (i.e., a funding source with many requirements) may add no more complexity than does a large number of stakeholders with a few requirements.

Risk

Using the logic in Lowrance (1976), Lewis (2006) defines simple risk as a function of three variables: threat, vulnerability, and damage. By replacing damage with Worth, Langford and Huynh (2007) capture risk through threat, vulnerability, and Worth. An element e of a system is associated with a risk, R_e , defined by

Equation 3. $$R_e = X_e * U_e * W_e = X_e * (1 - a_e) * W_e$$

where, * indicates the convolution that expresses the overlap and blending of factors; and where, threat, X_e , is a set of harmful events that could impact the element; vulnerability, U_e , is the probability that element e is degraded or fails in some specific way if attacked; Worth, $W_e = V[1 - Le]$, where L_e is the loss that results from a successful attack on element e; and susceptibility, a_e, is the likelihood that an asset will survive an attack. W_e is given by Equation 2. It may be a loss of productivity, casualties, loss of capital equipment, loss of time, or loss of dollars. Susceptibility is the complement of vulnerability. Equation 3 reflects these tentative affinities. One finds vulnerabilities in a worthy system from the threats to that system.

Since an element in a system (or network) may be connected to more than one element, the number of WTFs associated with the element is the degree of the element. Subscribing to Almannai and Lewis (2007), we obtain the system risk, R , as

Equation 4. $$R = \sum_{i=1}^{n+m} X_i * (1 - a_i) * g_i W_i$$

in which n denotes the number of elements; m denotes the number of links or WTFs, and g_i denotes the degree of connectedness (i.e., the number of connections) to the i^{th} element.

As a result of the WTF between two elements, e_1 and e_2, at the moment of their interaction, we have

Equation 5. $$\frac{W_{e_1}}{R_{e_1}} = \frac{W_{e_2}}{R_{e_2}}$$

It is the expression in Equation 4 that forms the basis for complexity management and acquisition.

Discussion

The approach extends the published and private works of Langford to identify and apply measurable objectives to characterizing and analyzing Gap Analysis. The two basic metrics are competitive *Worth* and a *Cost-to-risk* ratio. Both are displayable in an Enterprise Framework.

Gap Analysis fits into the overall scheme of acquisition by providing decision-makers with a structured and objective VoA from which to procure systems that satisfy defined needs. The desired results of Gap Analysis are to: (1) predict what we need for a postulated event, (2) compare what we need to what we have, (3) identify those items that need to be changed or added—along with the amount of investment in time and money required, and (4) enumerate the potential limitation of future capabilities. Recognizing there may be no means to maintain an optimal relation between the two limits—what we need and a potential limitation in capabilities—we assume the principles and practices of engineering are evolutionary and that the fundamental laws of physics prevail.

Further, we use generally accepted economics terminology, extended to encompass the notion that the price one pays for a product assumes and accounts for the loss realized to make the purchase (Taguchi, Chowdhury & Wu, 2005). That is, the purchase price of a product includes the cost of procurement—for example, the $1 purchase of a pen must be increased to $5 to include $0.50 of gasoline, plus amortized cost of maintenance, plus insurance, plus depreciation, and plus labor rate times travel time to drive to/from and make the purchase, etc. This notion states a willingness to spend (lose) $*x* to purchase a $1 item.

Following the accepted systems engineering process, product requirements are defined hierarchically, with each successive level offering greater detail via decomposition. However, unlike the different types of requirements that attach to various process models (e.g., functional and non-functional requirements), we define all processes and products by three measures—their functions, performances, and qualities (Langford, 2006; 2007). Relative to the Investment (Cost or its equivalent) to bring a product to operational capability, the product has determinable Worth. That Worth is expressed as a ratio of total value (i.e., operational capability or use as measured in terms of a unit of performance (e.g., work, throughput…) multiplied by Quality (effectively divided by the potential losses that could be incurred) and then divided by total lifecycle investment (i.e., expected cost for the use). As an example, if this ratio is less than 1, then the product has lower-than-expected worth.

Worth is related to both the vulnerabilities of the system and to the outputs of the system. The risks are a function of the threats and vulnerabilities, where threats are typed by magnitudes and frequencies, and vulnerabilities are determined by the likelihoods of success (DoD, 2006). The outputs of the system are related to the vulnerabilities through the price-demand elasticity curves (Lemarechal, 2001). The competition and the marketplace determine the threats; the operational strategy determines the vulnerabilities; and the triad of requirements determines the Worth (Langford, 2007).

To investigate the multivariate probability-density functions of the Risk Equation (Equation 3), a step-wise, two-variable analysis reveals both the boundary conditions and the relationships between Worth, Vulnerability, Threat, and Risk. Table 1 shows these boundary conditions. When any of the three variables (Worth, Vulnerability, or Threat) is zero, Risk is zero. And conversely, when Risk is zero, one or more of the three variables (Worth, Vulnerability, or Threat) is zero.

Table 1. Multivariate Boundary Conditions for Risk Equation

Worth = 0	Risk = 0
Worth = 0	Threat = ∞
Vulnerability = 0	Risk = 0
Threat = 0	Risk = 0
Risk = 0	Threat = 0
Risk = 0	Vulnerability = 0
Risk = 0	Worth = 0

A product that has Worth (quantified by the Worth Equation, Equation 2) from the developer’s point-of-view is one that has met the investment criteria for the desired set of functionality, performance, and quality requirements. From the user’s point-of-view, the Worth Function Equation emphasizes the trade that is made between the applicability of the purchased item (in terms of the item’s functionality, performance, and quality) and the total cost and time invested to purchase, use and sustain the product. This total cost and time (accumulated over the product’s lifecycle) is incurred not only during acquisition of the product, but also during the operation of the product and, finally, its disposal. This lifetime cost and time investment can also be viewed as a total loss to society (Taguichi, 1990), or as a specified loss as defined by a set of conditions.

Within the constraints of the boundary conditions indicated in Table 1, the relative ratios of Worth/Risk for an activity, a process, or a product or service may be displayed as probability density functions, and then summarized for display purposes as single data points. Figure 1 indicates two product lines—each drawn with designated points on curves depicting desirability, acceptability, and unacceptability. Product A (indicated on the upper right) has a higher market worth-to-risk ratio than Product B (lower left). The increasing worth-to-risk ratio moves generally upward. Product A can be compared to Product B on a one-to-one basis. Product parameters that indicate movement vertically upward reflect a decreasing threat but no change in vulnerability. Products that indicate movement horizontally to the left reflect decreasing vulnerability but no change in threat. Products that compete on price, such as the lower-priced Product B in Figure 1, have Event-space Strings (Langford, 2007) that are displaced upwards relative to their higher-priced competitors. Event-Space Strings are made up of sequences of causal events. These events are separated by probabilistic transitions rather than by either temporal or spatial (in the sense of being an adjacent event in a series) idealizations.

Consequently, Product B has a ”Desired” position, which is higher than that of Product A’s “Acceptable” position in the competitive Enterprise Framework. The higher position is indicative of the lower price (the lifetime cost to the consumer). If the lower price was offset by reduced functionality, performance, or quality, then the Worth would not increase. Product B is also located to the left of Product A, which indicates a reduction in vulnerability. This implies reduced risk and reduced threats. Therefore, as a competitive strategy, offering the lowest price with the highest utility is an efficacious strategy for competitors who are unable to match utility and pricing.

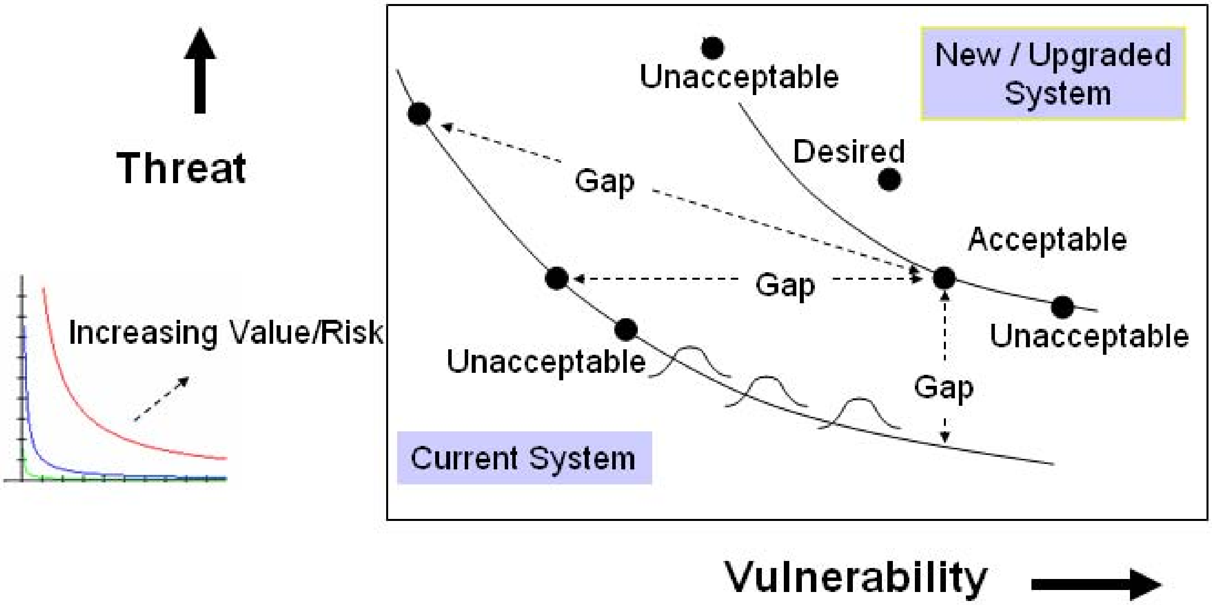


Figure 1. Enterprise Framework Illustrating the Worth-to-risk Assessment for Competing Products

The metrics for evaluating the Value-to-risk ratio or the Worth-to-risk ratio and their associated examples that describe contextual relationships are indicated in Table 2. These metrics are the rules which govern movement in the Enterprise Framework. Each rule corresponds to the impacts of business operations, competitive strategy, and the means or type of product offering. Event-space Strings are unique to a company's operations, strategy, and product offering. These rules describe the order of relative motions that have meaning appropriate to the context of a competitive space. Further, these rules are applicable to commercial products and services, the DoD battlespace, the procurement and acquisition landscapes, and the business environment considerations of business models and strategies.

In general, the Enterprise Framework is a visualization of decision-making processes in which the factors of value engineering, systems engineering, economics, acquisition, and operations research are involved. From such rules, the DoD Gap Analysis progresses in an orderly and logical fashion. Traditional statistical analysis, probability theory, and modeling are readily represented in proper context with conventional interpretation. As such, an error analysis results in confidence intervals for each point on the Event-space Strings. The scales of threat^{-1} and vulnerability^{-1} are determined by the probability of occurrence (0 to 1) multiplied by the frequency of occurrence (rate), and the odds of successfully inflicting loss (0 to 1), respectively. The vulnerability scale can be normalized in terms of dollars or numbers of items. Threats can be similarly normalized, as the situation warrants. Worth can be stated in either dollars or by numbers of items. Risk is a number between zero and one.

Of the possible rules (Table 2) for interpreting the Enterprise Framework, 31 have been identified; and thus far, 17 have been investigated. For example, Rule 1 implies a higher product utility (higher functionality, performance, and/or quality).

Table 2. Rules for Risk Equation in Enterprise Framework

Vulnerability ↓	Worth ↑	Risk -	Threat -	Rule 1
Vulnerability ↓	Threat ↑	Risk ↑	Worth ↑	Rule 2
Vulnerability ↓	Risk ↓	Threat -	Worth -	Rule 3
Vulnerability ↑	Risk ↑	Threat -	Vulnerability -	Rule 4
Vulnerability -	Threat ↓	Value -	Risk ↓	Rule 5
Vulnerability ↑	Worth ↓	Risk ↑	Threat -	Rule 6
Threat ↓	Risk ↓	Vul. -	Worth -	Rule 7
Threat ↓	Risk ↓	Unless vulnerability and/or worth ↑		Rule 8
Threat ↓	Risk ↓	or Worth ↓		Rule 8
Threat ↑	Risk ↑	Worth - and vulnerability -		Rule 9
Threat ↑	Risk ↑	Unless vulnerability and/or Worth ↓		Rule 10
Value ↓	Threat ↑	Risk -	Vulnerability -	Rule 11
Value ↓	Vul. ↑	Risk -	Threat -	Rule 12
Value ↓	Risk ↓	Threat -	Vulnerability -	Rule 13
Value ↑	Threat ↓	Risk -	Vulnerability -	Rule 14
Value ↑	Risk ↑	Risk -	Vulnerability -	Rule 15
Value ↓	Vul. ↑	Risk -	Threat ↑	Rule 16
Vulnerability ↓	Threat ↓	Worth ↑↑	Risk -	Rule 17

For a decrease in vulnerability (e.g., opening a new channel of distribution) due to a new competitive entrant in the marketplace (increase in threat), Rule 2 requires an increase in Worth commensurate with an increase in the Risk the enterprise is willing to accept. If the competitive landscape does not change and the enterprise's product remains the same, then opening up a new channel of distribution both decreases the product's vulnerability to competitive factors as well as reduces the enterprise risk with that product. Rule 4 indicates that an increase in vulnerability results in an equivalent increase in risk—if there are no changes in threat, and the product remains the same. Rule 5 indicates that a decrease in threat directly results in a decrease in risk if the enterprise's product and vulnerability are unchanged. Rule 6 implies that an increase in vulnerability decreases Worth (e.g., cost paid by a consumer increases due to a reduction in channel distribution) and increases the enterprise's risk if the threat landscape remains constant. Rule 7 indicates a reduction in the threat (e.g., competitors leave competitive space) results in commensurate reduction in risk, and the Worth and vulnerability stay the same. Rule 8 indicates a reduction in the threat (e.g., competitors leave competitive space) results in commensurate reduction in risk, unless either the product value or vulnerability increase—in which case, the overall risk would increase. Rule 9 indicates that as the threat increases, the risk increases, assuming the Worth and vulnerability remain constant. Rule 10 indicates that as the threat increases, the risk increases, unless either or both the value and vulnerability decrease. Rule 11 implies a greater investment in time and, therefore, a lower value and, hence, a higher risk. Rules 6, 12, and 16 each imply a lower product utility (insufficient functionality, performance, and/or quality). Rule 13 implies the product utility is worthless. Rule 14 implies a lower investment (time x money) and, therefore, a higher product Worth. Rule 15 implies the product has both higher utility and higher risk and is, therefore, worth more given that the threat and vulnerability remains unchanged. Rule 17 implies a disruptive technology or discontinuous innovation (Langford & Lim, 2007).

Following-up on this last rule that drives the display and interpretation of data in the Enterprise Framework, the discovery and analysis of potentially disruptive events and technologies (Rule 17) poses particularly incongruent issues for Gap Analysis. At issue is the degree of uncertainty that influences the choices and selection of alternatives in acquisition. The dangers of underestimating or not recognizing a disruptive technology or disruptive innovation result in a miscalculation of: (1) a sound operational vision, (2) the importance of planning and implementing an appropriate operational model, (3) the understanding of the relationships between current paradigms and a disruptive technology or innovation, and (4) the requisite acquisition strategy.

Finally, the graphical display of the competitive space (Enterprise Framework) found in Figure 1 portrays the results of Gap Analysis. From the view of the Enterprise Framework, the gaps reveal the needed capabilities, the prioritization of the capabilities, and the efficacy of the proposed alternatives. In the case of weapon systems, the Enterprise Framework is the geographical battlespace. It has physical structure, command structure, information structure, and engagement structure. Each structure is depicted in temporal- and event-driven layers. Truth is established through scenarios that illustrate capabilities that are enacted through these structures. Additionally, the Enterprise Framework illustrates opportunity shifts, allows evaluation of potential adversaries, and guides decision-makers' choices of what should be developed, indicates the system requirements that are satisfied by various strategies, illustrates potential target segmentations, and describes geographical arenas in the context of system capabilities.

General Formulation of Results

This work defines an Enterprise Framework in which to display the results of a Gap Analysis. For the purposes of this paper, an Enterprise Framework is a marriage of business parameters (reflecting operations and strategy) and product parameters (functions, performance, and qualities). The marriage is bonded through the structure of an expression of *Risk* (Equation 2).

In essence, the Enterprise Framework is an application framework that includes a multivariant view of a competitor's objectives, structure, and behavior. It is an adaptation of human activities into an abstraction that models the differences between these objectives, structures, and behaviors. Further, the framework is constrained by only two factors: geographical boundaries (for contextual structure) and a common event (to bring specificity to the nature of the competition).

Unlike the products of the domain analysis process, which processes imply a reference model for the semantics of the application domain, the Enterprise Framework described in this paper does not distinguish between such reference models, reference architectures, and the results of mapping a reference model (domain model) into an architecture style. Further, our Enterprise Framework is also not an analysis-only enterprise framework (Hafedh et al., 2002). It generally investigates the interfaces between a subject or action (e.g., issues, process or activity and other issues, processes or activities) and other subjects or actions.

However, in the case of Gap Analysis, our Framework has provided additional insight into its nature to examine its territory—the makeup of and changes in its surrounds, environs, relationships, and key drivers. There are different types of Gap Analysis "domains." Sometimes these domains are constrained by organizational demands, sometimes by personality issues, and sometimes by other circumstances. The internal structure of the Gap Analysis domains are arranged in particular patterns within an organization. Continuous functions (or patterns) are built and sustained by authoritative proclamation. Over time, such structures evolve to a mature environment that supports decision fitness and reliability in process planning. However, when the Gap Analysis territory is invoked, organized, structured, and enacted in response to stimuli, the outcomes of the work are predictably inconsistent and generally low in efficacy (Langford et al., 2006, December).

Additional questions arise when we formulate an overall strategy to analyze Gap Analysis: Do all organizations have Gap Analysis policies, strategy, procedures, processes, and rules? Are the enactments the same? How does Gap Analysis differ within and across organizations? What are the priority and process necessities that are observed? How and why does the position of Gap Analysis within an organization matter? Do the organization's position and priorities affect how Gap Analysis is performed, how it is interpreted, why it is done, how it is done, who does it, what is done, or when it is done? Are there general (or simple) rules that apply to all Gap Analyses?

These questions focus on the crux of the rules, roles (responsibilities), and mechanisms that determine how Gap Analysis is organized, and how host organizations change during the Gap Analysis process. It is one of the purposes of this research to move beyond the descriptive and correlative aspects of investigating Gap Analysis. While such an early mapping activity provides decision-makers the necessary framework to begin to understand and to identify areas for additional investigation, it must also identify the mechanisms responsible for the dynamics of Gap Analysis, and then determine how these mechanisms respond and contribute to the psychological cues, such as stimulation through signaling pathways.

The Enterprise Framework is a tool—a means to comprehend competitive business and operational models and product offerings, structured in forms that compare and definitively rate each. Additionally, market segmentations, niches, products, and upgrade strategies are readily apparent when coupled with a backward-looking series of event-space framings. An example of a generic Enterprise Framework is shown in Figure 1.

In the Framework, threat to the competitive offering is plotted versus its vulnerability. Risk is held constant, and Worth (Function, Performance, Quality divided by Investment) increases to the upper right. If a product is upgraded, the data point moves along the curve from the left to the right. The range of acceptability is indicated as Unacceptable (on the left) to Desirable (on the right). A Gap represents the space between data points. Moving from one curve to the next also indicates a Gap, but not an upgrade of an existing product. Rather, this Gap represents a form of Disruptive Technology in the competitive landscape. An increase in Worth is indicated as the points move "up" the curves to the top and to the left. A decrease in Worth is indicated as the points move "down" the curves to the bottom and to the right. The rules indicated in Table 2 illustrate the meanings and visualizations of Gaps. The scales of Threat^{-1} and Vulnerability^{-1} are relative scales for local normalization of the competitive parameters. In a more global summarization of Worth across multiple competitive domains, there are other issues, such as localized determination of value versus universal principles of value. For example, is it more valuable to go to a restaurant or to invest in a set of cooking utensils? Is it worth more to make such an investment? Some of the factors that need to be considered are the opportunities from "networking" at the restaurant versus the long-term investment in lowering the cost of eating. For the purpose of this paper, the authors relate only the localized competitive factors when comparing products.

Summary and Conclusions

Gap Analysis is fundamental to the US DoD acquisition system. The dismal results of time and cost overruns, ineffective use of constrained resources, and missed opportunities to make improvements without jeopardizing schedule and cost drive a critical evaluation of DoD acquisition (Rumsfeld, 2001). Since the cost and the success of acquisition are constrained by initial conditions, it is prudent to develop and apply tools that can help improve both the evaluation and the processes of acquisition. Gap Analysis, one of the key early-phase drivers of the acquisition process, has significant room for improvement.

This paper discusses the Systems Engineering Value Equation (Value Engineering) and the Worth Function in the context of the ratio of triadic decomposition of requirements based on functions, performance, and quality to the investment in time and cost. Investors and stakeholders have expectations about products they support. These expectations necessarily need to be met with a rigorous analysis of gaps. This notion has general adaptability and applicability to commercial and DoD acquisition. In the commercial sense, Gap Analysis tools can be used to better position products in the competitive market space. In the DoD sense, Gap Analysis tools provide a more effective use of constrained resources to support development activities.

The application of Gap Analysis to the general problem of satisfying requirements involves more than simply improving the methodology. Methodology that is encumbered with time-consuming steps and overburdened processes does not improve Gap Analysis. It is only through a streamlining of Gap Analysis that is efficacious, effective, and efficient that the forces and consequences of acquisition are better served. Thus, it is much more important and to the point to determine how to improve the outcomes of Gap Analysis (including the time to complete the Gap Analysis process) than to determine merely what can be improved with Gap Analysis. To that end, the actions of Gap Analysis should not be obstructed by insistence on unnecessary procedures and folderol. Straightforward application of the formulations laid out in this report result in the application of the sound value engineering and systems engineering processes that have generally become widely accepted as standard practices.

At least some future research on Gap Analysis should concentrate on the further expansion of the standards of earned value management as well as on the integration of new management practices to exploit fully the prowess of value engineering and systems engineering.

List of References

Almannai, W.I., & Lewis, T.G. (2007). Minimizing network risk and application to critical infrastructure protection. Preprint from *Journal of Information Warfare, 6*(2), 52-68.

Aristotle. (350 B.C.E.). *Physics*, Book II, Part 3.

Browning, T.R., Deyst, J.J., Eppinger, S.D., & Whitney, D.E. (2000, July). Complex system product development: adding value by creating information and reducing risk. In *Proceedings of the Tenth Annual International Symposium of INCOSE*. Minneapolis, MN: INCOSE.

Cantor, M. (2003, April). Thoughts on functional decomposition. *Rational Edge*. Retrieved August 4, 2007, from http://www.therationaledge.com/countentapr_03/f_functionalDecomp_mc.jsp

CJCS. (2005, May 11). *Operation of the joint capabilities integration and development system* (CJCSM 3170.01B). Washington, DC: Author.

CJCS. (2004, March 12). *Joint capabilities integration and development system* (CJCSI 3170.01D). Washington, DC: Author.

Draper, T. (1998). *Private communication.* Menlo Park, CA: Draper, Fisher, Jurvetson.

DoD. (2004, December 16). *Defense acquisition guidebook.* Chapter 1.3. Retrieved June 20, 2006, from http://akss.dau.mil/dag/.

DoD. (2006, August). *Risk management guide for DoD acquisition* (6th ed.). (Ver. 1.0). Washington, DC: Author.

Duzi, M. (2002). *Notional attitudes*. *Principles of logical analysis of language expressions and intensional logic* (Grant GACR: 401/03/1403). Czech Republic: VSB-Technical University of Ostrava.

Green, S. (1996). *Smart value management: A group decision support methodology for building design.* Unpublished doctoral dissertation. Department of Construction Management & Engineering, University of Reading, UK.

Hafedh, M., Fayad, M., Brugali, D., Hamu, D., & Dori, D. (2002). Enterprise frameworks: Issues and research directions. *Software Practice and Experience*, *32*(8), Number 8, 801-831 (DOI:10.1002/spe.460).

Homer, S., & Selman, A.L. (2001). *Computability and complexity theory.* New York: Springer-Verlag.

Jianjiang, C., Zhong, Y., Xiao, R., Sun, J. (2005). The research of decomposition-coordination method of multidisciplinary collaboration design optimization. *Engineering computations: International Journal for computer-Aided Engineering and Software, 22*(3), 274-285.

Kruchten, P. (2000). *The rational unified process* (2nd ed.). Boston, MA: Addison Wesley.

Kujawski, E. (2007). [Private communications with authors]. Naval Postgraduate School, Department of Systems Engineering.

Langford, G. (2006, April). Reducing risk of new business start-ups using rapid systems engineering. In *Proceedings of the Fourth Annual Conference on Systems Engineering Research* (Paper # 140). Los Angeles: University of Southern California and INCOSE, Los Angeles.

Langford, G. (2007, March). Reducing risk in designing new products using rapid systems engineering. In *Proceedings of the Asia-Pacific Systems Engineering Conference* (Paper # 18). Singapore: National University of Singapore and INCOSE, Singapore.

Langford, G., & Lim, H. (2007, March). Predicting and assessing disruptive technologies using event space modeling. In *Proceedings of the Asia-Pacific Systems Engineering Conference*, (Paper # 40). Singapore: National University of Singapore and INCOSE, Singapore.

Langford, G.O., Franck, R., Lewis, I., Huynh, T., Osmundson, J., & Sullivan, N. (2006, December). *Gap analysis.* Contract sponsored by the US Office of Secretary of Defense.

Langford, G., & Huynh, T. (2007, October). A methodology for managing complexity. In *Proceedings of the Systems Engineering Test and Evaluation—Complex Systems and Sustainability*. Sydney, AUS: Australian Government, Department of Defence, Defence materiel Organization, INCOSE, Sydney.

Lemarechal, C. (2001, January). Computational combinatorial optimization: Optimal or probably near-optimal solutions. *Lecture Notes in Computer Science*, *2241*, 112.

Lewis, T. (2006). *Critical infrastructure protection in homeland security.* Hoboken, NJ: John Wiley & Sons.

Li, M., & Vitanyi, P. (2006). *An introduction to Kolmogorov complexity and its applications.* New York: Springer Verlag.

Lowrance, W.W. (1976). *Of acceptable risk.* Los Altos, CA: William Kaufman.

Merton, R. (1938). Social structure and anomie. *American Sociological Review*, *3*, 672-682.

Miles, L. (1972). *Techniques for value analysis and engineering* (2nd ed.). New York: McGrawHill (first published 1961).

Pahl, G., & Beitz, W. (1995). *Engineering design: A systematic approach* (2nd ed.). New York: Springer-Verlag.

Poort, E., & Peter, D.W. (2004). Resolving requirement conflicts through non-functional decomposition. In *Proceedings of the Fourth Working IEEE/IFIP conference on Software Architecture* (WICSA'04) (pp. 11-14). Los Alamitos, CA: IEEE Press, 145-154.

Rumsfeld, D. (2001, September 10). *Bureaucracy to battlefield.* Remarks on DoD acquisition and logistics excellence week kickoff. Retrieved September 22, 2007, from http://www.defenselink.mil/speeches/2001/s20010910-secdef.html.

Sproles, N. (2002). Formulating measures of effectiveness. *Journal of Systems Engineering, 5*(4)*,* 253-263.

Taguchi, G. (1990). *Introduction to quality engineering.* Tokyo: Asian Productivity Organization.

Taguchi, G., Chowdhury, S., & Wu, Y. (2005). *Taguchi's quality engineering handbook.* Hoboken, NJ: John Wiley & Sons.

Von Boehm-Bawerk, E. (1909-1912). *Kapital and kapitalzins: Positive theorie des kapitales* (Stitch-sewn ed.)*.* Zweite Abteilung. London: Macmillan.

Wymore, W. (1993). *Model-based systems engineering.* Baca Raton, FL: CRC Press.

Panel 3 - Implications of Defense Industry Consolidation

Wednesday, May 14, 2008	**Panel 3 - Implications of Defense Industry Consolidation**
11:15 a.m. – 12:45 p.m.	**Chair:** **Mr. William C. Greenwalt**, Deputy Under Secretary of Defense for Industrial Policy, Office of the Under Secretary of Defense for Acquisition, Technology & Logistics **Discussant:** **Dr. Francois Melese**, Professor, Defense Resource Management Institute, Naval Postgraduate School **Papers:** ***Market Dominance, Efficiency, Innovation, and Globalization: A Case Study of the Tanker Competition between Boeing and Northrop Grumman/EADS*** **Dr. Nayantara Hensel**, Assistant Professor, Naval Postgraduate School ***The Changing Shape of the Defense Industry and Implications for Defense Acquisitions and Policy*** **Dr. Victoria A. Greenfield**, Visiting Professor, US Naval Academy

Chair: William "Bill" Greenwalt, as the Deputy Under Secretary of Defense for Industrial Policy, is a principal advisor to the Under Secretary of Defense Acquisition, Technology & Logistics (AT&L) and the Deputy Under Secretary of Defense Acquisition and Technology (A&T) on all matters relating to the defense industrial base. His office is responsible for ensuring Department of Defense (DoD) policies, procedures and actions stimulate and support vigorous competition and innovation in the industrial base supporting defense, as well as establish and sustain cost-effective industrial and technological capabilities that assure military readiness and superiority.

Prior to joining the DoD, Greenwalt was a Professional Staff Member of the Senate Armed Services Committee (Senator John Warner, Chairman) from March 1999 until March 2006, and was responsible for defense acquisition policy, industrial base, export control and management reform issues. Additionally, he served as deputy to the staff director and provided oversight and management direction of the committee's legislative activities. He was also a lead staff member for the Subcommittee on Readiness and Management Support and the Subcommittee on Strategic Forces. Previously, he served on the Senate Governmental Affairs Committee (Senator Fred Thompson, Chairman) as a Professional Staff Member responsible for federal management issues and committee press relations.

Greenwalt also served as a staff member for the Senate Subcommittee on Oversight of Government Management (Senator William Cohen, Chairman), where he was responsible for legislative efforts to reform federal information technology acquisition, culminating in the *Clinger-Cohen Act of 1996*. Prior to coming to the Senate in 1994, Greenwalt was a visiting fellow at the Center for Defense Economics, University of York, England; worked for the Immigration and

Naturalization Service in Frankfurt, Germany and served as an evaluator with the US General Accounting Office in Los Angeles, CA, where he specialized in defense acquisition issues.

Greenwalt graduated from California State University at Long Beach in 1982 with a Bachelor's Degree in Political Science and Economics and received his MA in Defense and Security Studies from the University of Southern California in 1989. He is married to Paula Mathews, and they live with their children, Geoffrey and Jenna, in Arlington, VA.

Discussant: Dr. Francois Melese earned his Bachelor's degree in Economics at UC Berkeley, his Master's degree at the University of British Columbia in Canada, and his Doctorate at the Catholic University of Louvain in Belgium. He was a research fellow at the Institut de Recherches Economiques et Sociales (IRES) and the Center for Operations Research and Econometrics (CORE) in Belgium before joining Auburn University's Business School in 1982. In 1987, he joined the faculty of the Naval Postgraduate School, and today is Professor of Economics at the Defense Resources Management Institute (DRMI). Besides teaching resident executive management courses for domestic and international government executives, he has taught short courses in over two dozen countries on public budgeting and defense management. He has consulted extensively, most recently for the Joint Chiefs of Staff and for the Deputy Secretary of Defense's Directorate for Organizational & Management Planning.

Melese has published over 50 articles and book chapters on a variety of topics in economics and management including: energy markets, labor and incentive systems, international trade, economic development, applied game theory, defense management, and public budgeting. At the request of NATO Headquarters and the State Department, he has represented the United States as an expert in defense management and public budgeting at NATO meetings in: Budapest, Hungary; Kyiv, Ukraine; Berlin, Germany; Garmisch, Germany; Yerevan, Armenia, and will be representing the US at the upcoming joint NATO/Marshall Center meetings in Ljubljana, Slovenia. His talk to the NABE will be based on a publication co-authored with Professor Jim Blandin and the Honorable Sean O'Keefe entitled, "A New Management Model for Government: Integrating Activity-Based Costing, the Balanced Scorecard and Total Quality Management with the Planning, Programming and Budgeting System" (2004, *International Public Management Review, 5*(2)). The paper can be retrieved from www.ipmr.net.

Market Dominance, Efficiency, Innovation, and Globalization: A Case Study of the Tanker Competition between Boeing and Northrop Grumman/EADS

Presenter: Dr. Nayantara Hensel is an Assistant Professor of Economics and Finance at the Graduate School of Business and Public Policy at the US Naval Postgraduate School. She received her BA (magna cum laude) from Harvard University where she was a member of Phi Beta Kappa. She received her MA and PhD from the Graduate School of Arts and Sciences at Harvard University. in Business Economics (Applied Economics). Prior to joining the faculty at the US Naval Postgraduate School, Dr. Hensel served as a Senior Manager at Ernst & Young, LLP and the chief economist for one of its units, was a Post-Doctoral Research Fellow at the National Bureau of Economic Research, taught at Harvard University and the Stern School of Business at NYU, and was an economist at NERA (part of Marsh & McLennan). Dr. Hensel's recent research has examined the impact of size and market structure on efficiency (economies of scale) in European and Japanese banks and on their tendency to open branches or merge, the impact of contracting arrangements and mergers on the utilization of railroad networks, the role of venture capital in DoD, the determinants of discount rates for military personnel, and the impact of online auctions on IPO pricing efficiency. She has published in a variety of journals, including: the *Review of Financial Economics,* the *International Journal of Managerial Finance,* the *European Financial Management Journal,* the *Journal of Financial Transformation, Business Economics,* and *Harvard Business School Working Knowledge.*

Abstract

The purpose of this analysis is to provide a case study of the competition between Boeing and Northrop Grumman/EADS for the Air Force refueling tankers contract and to discuss the role of many of these considerations in the controversy. This is an important case study because it highlights: (a) the concerns of the American people that they are continuing to lose manufacturing jobs overseas and the solutions that they are considering to lessen that problem; (b) the conflict between the concept of the US and European defense companies as partners against common threats to provide the best systems possible and the concept of them as competitors; (c) the concerns of an incumbent that it is losing its traditional edge; and (d) the desire to have an open and fair government procurement process in which all parties are able to accept the outcome that the process produces. This case study explores the background behind the contract, the reactions to the awarding of the contract, the reasons for the awarding of the contract, and the likely implications of the Boeing/Northrop Grumman-EADS competition for the competing firms, the government contracting process, and the global market.

1. Introduction

Over the past twenty years, following the end of the Cold War, the defense industrial base in the US has witnessed many changes. First, reductions in defense budgets during the 1990's contributed to consolidation among US defense contractors. Many defense industry sub-sectors manifested a 2/3 reduction in the number of prime contractors and came to be dominated by larger defense giants formed from the consolidations: Lockheed Martin, Boeing, Northrop Grumman, Raytheon, and General Dynamics. Second, the overall US economy witnessed an acceleration of the already apparent shift toward the services sector and away from the overall US industrial base in key manufacturing industries, such as steel and automobiles. As US manufacturing wages became globally uncompetitive, the corporate giants of an earlier era, burdened with generous pension plans and wage/benefit contracts with unions, went bankrupt. Third, the post 9/11 period has witnessed a broad range of security threats, including the emergence of a new type of threat in the form of terrorist groups. Many of these threats transcend the boundaries of nation-states and pose significant risks to all the members of the global community. Fourth, the new millennium has encouraged greater transparency and fairness in processes, ranging from corporate practices in the post-Enron world, to more up-to-date and open government procurement practices. These trends have resulted in the coalescence of the military forces of nation-states around the globe against these various security threats, including the threat of terrorism. Innovation continues to be important for the large US defense contractors as they compete with smaller entrants in a more open government procurement process, as they struggle against the concern that the US industrial base is shrinking overall and being replaced by overseas manufacturing, and as they handle the dual role of foreign companies as allies and as competitors.

The purpose of this analysis is to provide a case study of the competition between Boeing and Northrop Grumman/EADS for the Air Force refueling tankers contract and to discuss the role of many of these considerations in the controversy. This is an important case study because it highlights: (a) the concerns of the American people that they are continuing to lose manufacturing jobs overseas and the solutions that they are considering to lessen that problem; (b) the conflict between the concept of the US and European defense companies as partners against common threats to provide the best systems possible and the concept of them as competitors; (c) the concerns of an incumbent that it is losing its traditional edge; and (d) the desire to have an open and fair government procurement process in which all parties are able to accept the outcome that the process produces. This case study will explore the background behind the contract, the reactions to the awarding of the contract, the reasons for the awarding of the contract, and the likely implications of the Boeing/Northrop Grumman-EADS competition for the competing firms, the government contracting process, and the global market.

2. Prelude to the Announcement

During the past several years, recapitalization of the US Air Force has become an increasingly high priority. An important example of this imperative is the USAF’s need to upgrade its aerial refueling tankers. The average age of the existing KC-135 tankers is 47 years (Wolf & Shalai-Esa, 2008) and the planes were first put into service in 1957 (“Analysts,” 2008). The Air Force has 531 tankers from the Eisenhower period and 59 tankers built by McDonnell Douglas in the 1980’s (“Northrop group,” 2008), prior to its merger with Boeing (1997). Seeking to replace its ageing tanker fleet, the Air Force conducted a competition to award the initial $35 billion contract. Some have referred to the contract as “one of the largest military contracts in history” (Hinton, 2008b, March 11). This award was to constitute the first of three awards that could ultimately be worth $100 million (“Northrop group,” 2008; “Boeing to protest,” 2008), as the Air Force gradually replaces its existing 600-tanker fleet. The contract may involve the most expensive purchase in defense history, with the exception of the F-35 Joint Strike Fighter made by Lockheed Martin (Wolf & Shalai-Esa, 2008).

While there was some uncertainty over who the winner of the contract would be, many analysts thought that it would be Boeing because it had been providing refueling tankers to the USAF for almost 50 years and had, what was often referred to as a "monopoly." ("Northrop group," 2008). In an *Associated Press* article on February 22, 2008, it was reported, '"The incumbent is considered the favorite to win—an assumption already reflected in its stock price'" (Tessler, 2008, February 22). Indeed, the office of Texas Senator Kay Bailey Hutchison actually issued a statement on the morning of the announcement, February 29, 2008, (which it later retracted) that Boeing was the winner (Drawbaugh, 2008, February 29), while a poll of 10 industry analysts indicated that all of them were predicting a win by Boeing (Wolf & Shalai-Esa, 2008). Nevertheless, the Air Force did not release any hint of its decision prior to its announcement. Indeed, as of February 28, the day before the announcement, General Michael Moseley (Chief of Staff, USAF) noted that '"he himself did not know whether Boeing or Northrop Grumman would be awarded a potential $40 billion deal."' He stated, '"As you know by policy and law, I'm not in the acquisition business and have no idea which airplane I'm going to get"' (Wolf, 2008, February 28).

There was, however, some indication prior to the announcement, that the Air Force was concerned about a protest from the losing competitor. This could have been because the contract was so lucrative and important, and it felt that the loser would be disappointed. In addition, some officials may have anticipated that if Boeing, the incumbent tanker manufacturer, lost the contract, it would be more likely than Northrop Grumman or EADS to launch a protest. As early as February 22, it was reported that "the Air Force has said it expects a protest and has been extra careful in documenting its decision-making process." Lieutenant General Raymond Johns, the Air Force Deputy Chief of Staff for Strategic Plans and Programs, noted, '"We will not let politics dictate the best tanker for the Air Force'" (Hinton, 2008, February 22). Gen. Mosely continued, in his February 28 statement, that he hoped that whomever lost the contest would not challenge the result by lodging a protest with the GAO, which then has 100 days to make a recommendation as to whether the contract competition should be re-opened. His observation reflected concern about delaying the time line for the delivery of the tankers to the USAF (Wolf, 2008, February 28).

3. The Announcement

On February 29, 2008, after the markets closed, the Air Force announced that the Northrop Grumman/EADS bid for the aerial refueling tanker had won the contract (Wolf, 2008, February 29). As mentioned earlier, this comprised the first of three awards that could ultimately be worth $100 billion ("Northrop group," 2008; "Boeing to protest," 2008), although the winner of this competition would not necessarily be the winner of the subsequent competitions (Wolf & Shalai-Esa, 2008). The contract awarded was actually worth $1.5 billion, covering 4 test aircraft. The intent was then to buy 175 more planes, for a total value of $35 billion. The Air Force hoped to operate the new tankers in 2013 (Wolf & Shalai-Esa, 2008). While the $35 billion amount would stretch over 10-15 years, an additional $60 billion in revenue could come from maintenance and parts (Hinton, 2008b, March 11).

The tanker in the winning bid, the KC-45, was a modification of the Airbus A330 (Hepher, 2008, March 3). Air Force General Arthur Lichte noted that the KC-45A, provided "More passengers, more cargo, more fuel to offload" and that the bigger capacity of that tanker had been an important consideration in awarding the contract ("Northrop group," 2008). The Northrop tanker carried more fuel—250,000 pounds—than the Boeing tanker at 202,000 pounds ("Tanker Deal," 2008). Finally, Loren Thompson at Lexington Institute, was quoted as observing that, "With Northrop, the military could have '49 superior tankers operating by 2013' […] while Boeing's proposal would give it 'only 19 considerably less capable planes'" ("Tanker Deal," 2008).

4. Reaction to the Announcement and the Differences in the Two Bids

Almost immediately following the announcement that its bid had not been selected, Boeing indicated that it was upset at the decision. On Friday, February 29, following the award of the contract, Boeing released an announcement stating, "We believe that we offered the Air Force the best value and the lowest risk tanker for its mission. Our next step is to request and receive a debrief from the Air Force" ("Analysts," 2008). Boeing noted that it would not decide on whether to formally appeal the contract decision until after the Air Force had briefed them on why the contract had been awarded to the Northrop/EADS team ("Northrop group," 2008). On Tuesday, March 4, the Air Force agreed to provide a briefing sooner to Boeing after Boeing had alleged that delaying a briefing until March 12 would be '"inconsistent with procurement practices."'

In its public press release requesting an immediate briefing on the tanker, Boeing argued:

> "based on values disclosed in the Air Force press conference and press release, the Boeing bid, comprising development and all production airplane costs, would appear less than the competitor. In addition, because of the lower fuel burn of the 767, we can only assume our offering was more cost effective from a life cycle standpoint. […] Initial reports have also indicated that we were judged the higher risk offering […] Northrop and EADS are two companies that will be working together for the first time on a tanker, on an airplane they've never built before, under multiple management structures, across cultural, language, and geographic divides […] Initial reports also indicate there may well have been factors beyond those stated in the RFP, or weighted differently than we understood they would be, used to make the decision" ("Boeing Requests," 2008).

On March 5, Jim Albaugh, CEO of Boeing's Integrated Defense Systems, argued that Boeing had provided the Air Force exactly what was requested in its RFP and for a lower amount than the $35 billion price indicated (Carpenter, 2008). In response to General Lichte's comment that the greater size of the Northrop-EADS tanker was important in the decision-making process, Albaugh argued that, '"In our reading of the RFP, it wasn't about a big airplane. If they'd wanted a big airplane, obviously we could have offered the 777. And we were discouraged from offering the 777"' (Carpenter, 2008).

On Friday, March 7, Boeing met with the Air Force to receive its briefing on why it lost the contract (Palmer, 2008). After the meeting, Boeing stated that it was "'seriously considering'" launching a protest ("Boeing: Far," 2008). While the Air Force had said that the Northrop Grumman/EADS bid did better than the Boeing bid on four of the five criteria, Boeing claimed that it scored marks which were identical to those of Northrop/EADS on the five main criteria (Rigby, 2008, March 11). John Young from the Pentagon reiterated that there were “substantial capability and cost differences” between the two proposals (Rigby, 2008, March 11). Following the briefing, Boeing had 10 days to file a protest with the GAO. Then, the GAO would have 100 days to determine if the contract had been awarded fairly or if a new competition would be needed (Wolf, 2008, March 7).

On Monday, March 10, Boeing announced that it would challenge the decision (“Boeing to challenge,” 2008). Boeing argued that the Air Force had changed its requirements on the amount of ramp space and how closely the tankers could be parked to each other and that ‘“the changes were designed to keep them [Northrop] in the competition”’ (Hinton, 2008a, March 11). Boeing felt that the process was ‘“replete with irregularities,”’ which ‘“placed Boeing at a competitive disadvantage”’ and that ‘“the original mission for these tankers—that is a medium-sized tanker where cargo and passenger transport was a secondary consideration—became lost in the process, and the Air Force ended up with an oversized tanker.”’ Mark McGraw, manager of Boeing’s tanker programs, stated, ‘“As the requirements were changed to accommodate the bigger, less capable Airbus plane, evaluators arbitrarily discounted the significant strengths of the KC-767, compromising operational capabilities, including the ability to refuel a more versatile array of aircraft such as the V-22 and even the survivability of the tanker during the most dangerous missions it would encounter”’ (“Boeing Protests,” 2008). McGraw did not think that Boeing had made a mistake in this competition and stated, ‘“Last year we won nine out of 11 major competitions we went after. I think we know how to win competitions’” (Wingfield, 2008).

On March 11, Secretary of the Air Force Michael Wynn stated that the Air Force did not steer Boeing from proposing a larger plane and that ‘“these are competent suppliers. They can read a proposal’” (Rigby, 2008, March 11). Late on Tuesday, March 11, the Air Force stated that this decision gave ‘“the best value to the American taxpayer and to the warfighter’” and that it had continually provided the bidders with feedback on their proposals to ‘“provide transparency, maintain integrity, and promote fair competition,”’ while suggesting that the larger size of the Northrop/ EADS tanker was very much a deciding factor (Tessler, 2008, March 11). Nevertheless, the 767 model had some advantages over the Airbus 330-200 model. The Boeing tanker could land on narrower, shorter airstrips, such as those in developing countries in Africa, or in Afghanistan (Hinton, 2008, February 22).

One of the concerns cited by critics of the Northrop/EADS proposed tanker design is that they are larger and will require more fuel (Shalal-Esa, 2008), which will be problematic with increases in fuel prices. On March 17, Boeing released a report stating that, over the next 40 years, it would cost the Air Force an extra $30 billion in fuel costs to operate the 179 Airbus A330-200 refueling tankers relative to a similar number of Boeing tankers. The A330-200 requires 24% more fuel than the 767-200ER. At $100 per barrel for oil, the Airbus fleet would cost the Air Force $25 billion more in fuel costs over 40 years, while at $125 per barrel, it would be $29.8 billion more. At Boeing’s briefing, the Air Force did note “that they placed little value on fuel and maintenance lifecycle costs” (“Boeing Study,” 2008).

Will Boeing’s protest succeed? As of this writing (late March, 2008), the evidence suggests that it may not, but that the protest itself may delay the Air Force’s timeline for obtaining the new tankers. Analysts, such as George Shapiro at Citigroup, have argued that Northrop/EADS will end up keeping the contract, but that the dispute will take 6-9 months to resolve (Hinton, 2008a, March 11). Myles Walton, an analyst at Oppenheimer & Co., stated, '"given the initial judgment by the Air Force combined with the Northrop team's better score on four out of five criteria, we anticipate Boeing's protest will be denied"' (Rigby, 2008, March 11). On March 18, Mark McGraw, the tanker manager for Boeing, stated '"We know its an uphill battle"' and that '"I think the best we can hope for is another shot'"—perhaps a portion of the competition being re-run"' (Wolf, 2008, March 18). Northrop's tanker manager, Paul Meyer rated the chance of the GAO upholding Boeing's protest as '"low"' (2008, March 18).

Complaints are often unsuccessful with the GAO. Only 249 of the 1327 bid complaints lodged with the GAO in 2006 received an official decision; in 71% of those, the GAO denied the complaint and supported the government's earlier decision ("Boeing to protest," 2008). In fiscal year 2007, of the 1393 cases filed and closed, 16% of them were ruled to have merit by the GAO (Crown & Epstein, 2008). Boeing has, however, been involved as the losing party in some of the GAO decisions recently. In 2007, Lockheed Martin and Sikorsky Aircraft (part of United Technologies) successfully protested the awarding of a contract to Boeing for a $15 billion helicopter. In February, 2008, the GAO recommended that the award of a $1.2 billion contract for airplane maintenance to Boeing should be re-examined by the Air Force, which, in turn, has agreed to reevaluate it (Rigby, 2008, March 3).

Boeing is, in many ways, behaving like a traditional, incumbent corporate giant who is upset that its traditional turf is being encroached upon. Many of its arguments, discussed previously, have focused on the fact that they did not understand the Air Force's preferences, and that, consequently, they did not provide a more innovative model of tanker. Last year, Boeing only sold 36 Boeing 767's—a variation of which was proposed by Boeing for the tanker competition last year—and, having sold 1000 over the past 30 years, only has 51 left to deliver. This suggests that, in the absence of additional orders, the 767 assembly lines near Seattle may close down. The 787 Dreamliner, on the other hand, which is a successor to the 767, received 369 orders last year (Rigby, 2008, March 3). Boeing argued that, '"To some extent, the requirements [of the Air Force] steered us to the 767"' (Vorman & Wolf, 2008). EADS, on the other hand, read the same RFP as Boeing, yet proposed a more innovative model of tanker, particularly in designing a new boom. Indeed, on March 4, EADS confirmed that it had completed the first test of the Air Refueling Boom System for the aircraft ("EADS confirms," 2008).

Boeing has had a previously difficult history with Air Force tankers. In 2004, Congress voted to overturn of the USAF plan to lease and buy 100 modified KC-767 tankers from Boeing for $23.5 billion following a Pentagon procurement scandal, in which one of the key Air Force procurement officials, Darleen Druyen, and the CFO of Boeing, Michael Sears, went to jail. The scandal was brought to light partially with the assistance of Senator McCain's office (Wolf & Shalal-Esa, 2008). It is unclear whether this in any way impacted the decision, other than that the prior history of scandal encouraged the Air Force to make this procurement decision very transparent and well-documented and that the scandal delayed the Air Force's strategy of replacing its aging tanker fleet.

Boeing has had a history of tardiness and delays, which reduces its argument that it is a reliable supplier. For example, it delivered its first tanker to Japan in late

February, 2008, when the original target date had been in 2005. It is two years behind schedule with Italy, and hopes to deliver the first of four tankers to it this year (Rigby, 2008, March 3). Furthermore, Boeing has experienced delays on the 787 Dreamliner, which has led to a decline in its stock price since the summer of 2007 (2008, March 3). Indeed, before the contract results were announced, on February 21, Japan Air Lines, one of Boeing's best customers, announced that it was considering buying some Airbus A350 XWR planes due to the production delays for the Boeing 787's. Indeed, due to the lateness of the planes, some airlines, such as Air India and Qantas, have stated that they are likely to seek financial compensation from Boeing (Tessler, 2008, February 22). News on delays continued to be announced after the awarding of the contract—on March 12, it was reported that Boeing may not complete more than 45 of its 787's by next year, which is a change from its previous plan, which had involved the delivery of 109 planes next year (Hinton, 2008, March 12).

5. Should the Contract be Awarded to a Foreign Contractor?

One of the central concerns surrounding the awarding of the contract was that Boeing, an American firm, had lost its bid to a contracting team which involved a foreign contractor. This concern embodied several issues: (a) the possibility that US defense jobs were being lost to the European defense sector; (b) concerns that systems key to national security would be made by a foreign contractor; and (c) an overall fear that the US manufacturing industry is shrinking and the economy is shifting toward services. Indeed, in the official press releases, Northrop Grumman was referred to as the winner of the contract, while the role of EADS was downplayed (Morgan, 2008).

The Congressional representatives from the regions in Washington, Kansas, and Connecticut that would have benefited if Boeing had received the contract have strongly protested the decision. The Congressmen from the Seattle area claimed to be "outraged," while Kansas Representative Todd Tiahrt stated that he would seek a review of the contract decision (Drawbaugh, 2008, February 29). On Monday, March 3, a group of lawmakers from Kansas and Washington wrote to Defense Secretary Robert Gates and asked that the Air Force explain to Boeing why it lost the contract rather than wait until mid-March to do so (Drawbaugh, 2008, March 3). On March 5, members of the congressional delegation from Connecticut formally requested a briefing on why Boeing had lost the contract. Their concern was linked to the fact that the engines for the Boeing tanker would have been made by Pratt & Whitney, based in East Hartford, Connecticut and the electrical systems would have been made by Hamilton Sundstrand in Windsor Locks, Connecticut ("Conn," 2008). On March 7, the Kansas Senate adopted a resolution with a unanimous vote asking the President and Congress to block the contract ("Kan. Senate," 2008). On March 11, Representative Todd Tiahrt of Kansas announced that he was developing a bill to block funding for the Northrop tanker (Drawbaugh, 2008, March 11).

On the other hand, the Northrop Grumman/EADS tanker would create jobs in the US, especially in Alabama, and the Alabama Congressional delegation was very supportive of the results. Senator Richard Shelby (Alabama) noted that the contract would bring 7,000 jobs to Alabama (Drawbaugh, 2008, February 29) and that '"Any assertion that this award outsources jobs to France is simply false"' (Drawbaugh, 2008, March 3). Senator Jeff Sessions (Alabama) noted, '"In reality, what we're talking about is the insourcing, into America, of an aircraft production center that would bring 2500 jobs to our area and 5,000 to our state"' (2008, March 3). Kansas Representative Tiahrt, on the other hand, stated, '"I cannot believe we would create French jobs in place of Kansas jobs,"' while a joint statement of lawmakers protesting the decision noted, '"We are outraged that this decision taps European Airbus and its foreign workers to provide a tanker to our American military"' (2008, March 3).

Actually, both the Boeing tanker and the Northrop/EADS tanker would create jobs domestically and overseas. About 85% of Boeing's tanker would have been made in the US. Boeing argued that 44,000 new and existing jobs would have been assisted by the contract, across 40 states and 300 suppliers. Wichita, Kansas and Everett, Washington would have been major locations for tanker production, and the engines in the tanker, made by Pratt & Whitney, would have been made in Connecticut. Nevertheless, some portions of its tanker would have been made overseas—the tail in Italy and the fuselage in Japan (Tessler, 2008, March 6).

About 60% of the Northrop/EADS tanker would be made in the United States. This tanker was originally projected to create 25,000 jobs nationwide, including several thousand jobs in Mobile, Alabama, where the final assembly work was to take place (Vorman & Wolf, 2008). On March 10, however, Northrop's estimate of jobs created doubled to 48,000 jobs because it used more recent data and a formula from the Dept. of Labor in forecasting jobs in the aerospace industry (Crawley, McSherry, Rigby & Vorman, 2008). This estimate topped Boeing's 44,000 jobs. General Electric would build the engines for the Northrop/EADS tankers in North Carolina and Ohio ("Northrop group," 2008) and expected to make $5 billion from the contract (Witkowski, 2008). The contract would, however, also assist the European defense industry. The wings would be manufactured in the UK, such that 9,000 jobs would be created. GE Aerospace's British arm would also be involved (Lagorce, 2008, March 3). The Airbus-330, of which the KC-45 is a modification would have parts made in Germany, France, Spain, and Great Britain, but assembly of the KC-45 would occur in Mobile, AL (Wolf & Shalal-Esa, 2008). While Northrop argued that the contract would result in 2,000 jobs shifting to the US from Europe, EADS argued that the assembly plants in Mobile would result in the creation of new jobs in the US, not in jobs moving from Europe to the US ("Northrop Grumman," 2008).

Labor unions in the US were concerned that the Air Force did not consider US jobs when it awarded the contract, and that EADS received subsidies from European governments for years, creating a playing field which is not level. On March 3, the Association of Machinists and Aerospace Workers requested Congress enact legislation to prevent the US from awarding contracts to overseas companies receiving government subsidies, since, in a complaint filed by the US Trade Representative, the EU had been accused of providing subsidies to Airbus which are anticompetitive (Vandore, 2008, March 3). Airbus CEO Tom Enders, in response to criticism that Airbus was destroying more American jobs due to its subsidies than it could create by building tankers in the US, noted that it sourced $11 billion from the US for Airbus and has been the largest single customer outside the US for the aerospace industry (Hepher, 2008, March 7).

The AFL-CIO and the United Steelworkers Union also echoed concerns about sourcing the contract to a foreign manufacturer (Shalal-Esa, 2008). In a statement reported on March 3, the General Vice President of the International Association of Machinists and Aerospace Workers, Rich Michalski, said, '"President Bush and his administration have today denied real economic stimulus to the American people and chosen instead to create jobs in Toulouse, France"' ("Boeing calls," 2008). Senator Hillary Clinton from New York stated that "she found it 'troubling the government would decide to award the contract to a team including a European firm it is simultaneously suing at the World Trade Organization for receiving illegal subsidies"' (Lagorce, 2008, March 3). Senator Barak Obama from Illinois was also concerned that Boeing, based in Chicago, had lost the contract (Daly, 2008). Similarly, French unions protested the loss of assembly jobs to the US, since the tankers would be assembled in Mobile, AL ("EADS confirms," 2008).

Debates concerning job creation and destruction, similar to those in the tanker controversy, have occurred in a variety of different US manufacturing industries over the past twenty years and have focused on the broader issue of whether the US should get the best product at the lowest cost or artificially try to prop up uncompetitive industries. Senator McCain noted, '"I've never believed that defense programs should be—that the major reason for them should be to create jobs. I've always felt that the best thing to do is to create the best weapon system we can at cost to taxpayers'" (Drawbaugh, 2008, March 3). These thoughts were echoed in the comments of Pentagon acquisition chief John Young, who noted, '"I don't think anybody wants to run the department as a jobs program,"' further arguing that lawmakers usually focused on asking him to reduce the costs of weapons systems, and that a decision by Congress to ban sourcing of contracts to foreign companies could lead to reciprocal retaliation on the part of the Europeans (Shalal-Esa, 2008). Defense Secretary Robert Gates stated that '"defense manufacturing was a global business"' and that '"we sell aircraft and ships and weapons systems all over the world. The four countries that I just visited in Asia and in the Middle East—Australia, Indonesia, India, and Turkey—all have an interest in acquiring American aircraft, as an example"' ("Northrop Grumman," 2008).

The preceding comments reflect an awareness of the defense industry as a global industry in which the US, Europe, and other countries need to unite to combat global threats at various levels, including the terrorist threat. The growing interconnectedness between various countries is evident across a variety of other industries in our global economy. Furthermore, Boeing itself is an example of a global firm in that it makes weapons systems for other countries, so its hard for it to argue that it is unfair for a government to outsource a contract to a foreign supplier. Boeing sells C-17 planes to the UK, Australia, and Canada; F-15 jets to Japan, Korea, and Singapore; and aerial refueling tankers to Italy and Japan. Of the $66.4 billion comprising Boeing's 2007 revenue, about $27.1 billion came from overseas sales (commercial and military). Sales to Europe comprised $6.3 billion, of which 16% of that came from sales to the military (Tessler, 2008, March 6). Overall, about 13% of Boeing's total revenues from defense production came from overseas and included contracts to produce rockets in France, "early-warning" systems in South Korea and Turkey, and helicopters in Saudi Arabia, Israel, and Egypt. About 5% of Northrop Grumman's revenues in the defense arena came from contracts with other countries (Wingfield, 2008).

6. Implications of the Contract Award

6.1 Toward Greater Global Cooperation?

The award of the tanker contract to a team which includes a foreign contractor is indicative of the recognition of the importance of forming a global effort with the most innovative products against a variety of immediate and long-term actual and potential threats. Hopefully, the protests against the awarding of the contract will not disrupt the overall message of global cooperation. French President Nicholas Sarkozy stated on March 3, '"If Germany and France had not shown from the beginning that we were friends and allies of the United States, would it have been possible to have such a commercial victory?"'(Hepher, 2008, March 3). Significantly, EADS failed in a similar competition in 2003, at the time when the then-President of France, Jacques Chirac, was opposing the involvement in Iraq. Sarkozy, on the other hand, has established closer ties with the US, as evidenced by France's support of the US position on Iran's nuclear activities (Hepher, 2008, March 5). Consequently, while the victory of the Northrop/EADS team was based on the perception that their product was better, it may have been assisted by greater US-French cooperation, and the award of the tanker contract will reinforce and enhance that cooperation.

Although the popular press has noted that concerns about national security could be key in involving a foreign contractor in manufacturing US weapons systems, there have been other instances in which foreign contractors have worked on key US defense projects. For example, EADS has been working on a $2 billion Army contract for two years to replace 345 "Huey" helicopters, in addition to providing the Coast Guard with radar systems and search and rescue aircraft (Wingfield, 2008). The presidential helicopter was partially built by Italy's Finmeccanica, while Britain's BAE systems has been involved in a number of US DoD projects, since it purchased United Defense Industries in 2005 (Lagorce, 2008, March 3). Significantly, Boeing did not discuss national security issues in its formal protest (Wingfield, 2008), possibly for this reason. Moreover, all classified military technology on the KC-45a would be installed by Northrop after the aircraft was assembled, so that EADS would not be handling it (Hinton, 2008, March 10).

Furthermore, to alleviate national security concerns about having EADS as a partner on the contract, Germany and France are legislating changes to EADS' corporate charter preventing foreign investors, such as Russian or Middle Eastern shareholders, from obtaining large stakes in the company. The plan will give the Germans and the French a golden share, so that they can block stakes over 15%, or they can provide EADS with a poison pill (Lagorce, 2008, March 7).

Although Boeing has painted the conflict as a competition between a US company and a European one, much of its concern is that of a traditional incumbent watching its competitor and arch-rival, Airbus, encroach on one of its key contracts. This is not the first time that Boeing and EADS have competed over a tanker contract, and that EADS has won. Since 2001, Boeing and EADS have faced each other in competitions for tankers in six countries, of which EADS has won four of the competitions to supply a total of 25 planes (Saudi Arabia, UAE, Australia, and Britain), while Boeing has won in Italy and Japan to supply 8 planes ("Boeing's trouble," 2008).

The reasons why EADS has triumphed in some of the other competitions is that, in recognition of a global marketplace, the contract was awarded to the bidder which seemed the most sensitive to the needs of the client, the most flexible, and the most willing to make investments in the relationship. EADS, as a newcomer in the tanker business, has manifested the traditional behavior of a successful entrant in terms of being innovative and absorbing risk, while Boeing has played the role of the traditional incumbent. For example, Boeing and BAE Systems in the UK competed against EADS for a $26 billion contract to replace the UK's fleet of military refueling tankers in 2004, and lost the contract to EADS, which had not built a tanker before, and which had proposed a modification of the commercial Airbus A330. The UK felt that EADS was more willing to make concessions and assume the financial risk in constructing the planes and then leasing them to the government, whereas Boeing did not offer such terms. Although Boeing's C-17 transport plane had been successful there, the tanker business had been handled by a different division of Boeing than the C-17. The competition in Australia provides another illustrative example in that the Australian government was impressed by EADS' willingness to use its own R&D money to develop and test a boom, whereas Boeing used an older boom in its proposal, and suggested that it would build a newer type of boom only if it won the large US contract ("Boeing's trouble," 2008).

6.2 Impact on Boeing

Analysts have suggested Boeing's loss of the contract to the Northrop Grumman/EADS team "is part of a gradual erosion in Boeing's defense operations" and that this loss, combined with the reputational loss from the earlier tanker procurement debacle in 2004, is not helpful to its image (Rigby, 2008, March 3). Some analysts, such as John Kutler, have noted that, '"I thought, for a number of reasons, it would be difficult for the Air Force to pick Boeing,'" arguing that when Rumsfeld in 2006 jettisoned the plans to lease Boeing 767's, change might be in the wind ("Analysts," 2008). Furthermore, Boeing's delays on the 787's have not provided it with the aura of a reliable supplier. On the other hand, EADS may face delays, and only time will tell if that will occur, since its contracts with UAE and Saudi Arabia were signed last year; the Australian tankers aren't due until 2009, and the British tankers aren't due until 2011 ("Boeing's Trouble," 2008).

Recently, several of Boeing's contracts have run into problems. The "virtual fence" project along the border between US and Mexico has been pushed back three years to 2011 due to technical problems, and the company has spent twice the amount of the $20 million contract to fix these problems. The GAO, in February, found that three contracts with Boeing cost the government $3 billion more, with cost overruns of as much as 30% coming from higher expenses in labor and materials (Caterinicchia & Tessler, 2008).

Will Boeing's loss of its traditional role in building USAF tankers put it at a disadvantage in other competitions for tankers domestically and abroad? This is possible, given that EADS' tankers, as discussed previously, have been chosen over Boeing's tanker in several other competitions. If this trend continues, and if Japan and Italy may end up with "orphan fleets"—i.e., they are the only countries with Boeing tankers—then these fleets may cost more to maintain than if Boeing had developed the scale economies in costs to maintain the parts through obtaining other contracts, especially the US contract. As a result, other potential customers may be less likely to choose Boeing in the competition when they see these higher maintenance costs, and the cycle will become self-reinforcing ("Boeing's Trouble," 2008).

The loss of the tanker contract in itself should not affect Boeing on an annual basis, in that it only would have led to 12-18 additional tankers per year, which is a small number in comparison to the 450 commercial aircraft that it makes each year (Tessler, 2008, March 6). But, since it is a very large contract in the long-run, at a time when defense expenditures could plateau, it could have long-range significance.

On the other hand, even while the tanker contract announcement and protests were at their peak in late February through mid-March, 2008, Boeing was still announcing new orders and the award of new contracts. For example, on February 25, 2008, before the contract decision on the tankers became public, Boeing won a $77 million contract with the Air Force to install 37 infrared, anti-missile systems (“Boeing Wins $77M,” 2008). On March 5, it was reported that Boeing and Bell Helicopter (part of Textron) had won the contract to provide the V-22 Osprey with spare parts—a $204.5 million Navy contract (“Boeing, Bell,” 2008). On March 13, Boeing won a $32.8 million contract to provide the Air Force with Combat Survivor Evader Locator radio systems (“Boeing wins $32.8M,” 2008). On March 14, Boeing won a $28.2 million contract to provide the Navy with parts for the Growler attack aircraft (“Boeing wins $28.2M,” 2008). Over the preceding week, Boeing also listed orders for 85 new planes (“Boeing shares,” 2008). Finally, on March 17, Raytheon and Boeing won $89.5 million worth of contracts to provide radar systems to the Air National Guard and to the Air Force (“Raytheon,” 2008).

The key to Boeing’s long-run success is its ability to innovate and to be willing to modify its products to the needs of the customer. Merely satisfying the stated requirements of the customer is not always the best strategy in an increasingly competitive global marketplace, with many new entrants. Moreover, Boeing needs to continue to invest in assets specific to its relationship with customers. Hopefully, its protest on the awarding of the tanker contract to Northrop/EADS and the resulting delay in the Air Force’s time trajectory in obtaining new tankers will not reflect negatively on its long-standing relationship with the Air Force. Nevertheless, Boeing needs to focus on the lessons from its loss in this competition and other competitions, rather than expending significant energy combating the outcome of those earlier decisions.

6.3 Impact on EADS

The award of the highly publicized tanker contract to the Northrop Grumman /EADS team will provide EADS with a much-needed boost. Financially, it has been struggling for several reasons: (a) the weak dollar—EADS airliners sell in dollars, but often pays its suppliers in euros (Hepher, 2008, March 3); (b) the financial impact of its delays with the A-380 and, recently, with the A400M (Lagorce, 2008, March 3), which was supposed to debut in July and which has been delayed until October (Hepher, 2008, March 7); and (c) some flattening in customer expenditures relative to previous years. On March 11, 2008, the reported losses for 2007 were worse than expected. Its net loss was 446 million euros in 2007 (the net loss was forecast at 329 million euros) and represents a deterioration from its 99 million euro profit in 2006. The rise of the euro reduced the revenue at Airbus by $1 billion (Hepher, 2008, March 11).

EADS has been delighted at the award of the contract, partially because the contract will provide it with a greater capability to penetrate the US defense market and possibly to position it better to win future contracts. The existing EADS aerial refueling tanker has already won competitions in Australia, Saudi Arabia, the UAE, and Britain (Hepher, 2008, March 3), and its success in the US marketplace against an established competitor will help it to gain greater traction. Indeed, on March 27, 2008, a consortium led by EADS (and also including Rolls Royce, Cobham and Thales, and the VT Group) won a $26 billion contract in the UK over 27 years to provide 14 Airbus 330-200 tankers to replace the RAF's ageing fleet of VC-10 and Tristar refueling tankers. This contract had been under negotiation since 2004, and the success of the Northrop/EADS bid in the US may have helped to generate positive momentum (Pfeifer, 2008). EADS' success in both the US contest and the UK contest vindicates its strategy to increase its defense capabilities and not to depend entirely on commercial programs. Moreover, this KC-45a contract, combined with the weak dollar, may lead to EADS making an acquisition in the US (Vandore, 2008, March 10), to obtain an even greater foothold in the US market.

6.4 Implications for the Government Procurement Process

The award of the tanker to the Northrop/EADS consortium suggests that the government procurement process does not always favor incumbents and that there is an increasing emphasis on obtaining the most appropriate product at the best cost. Furthermore, it continues the precedent of transparent, open processes—which are often open to the global marketplace, especially when the range of national security threats, such as the terrorist threat, is more globally focused.

One concern is the potential impact on the government procurement process if the tanker's funding is successfully blocked by Congressional representatives who did not support the decision to award the contract to Northrop/EADS. If this occurs, it will send a signal that the political landscape—factors such as which states benefit from the award of a given contract and which Congressional representatives have greater power—can overturn a decision made by defense procurement experts who are weighing cost and quality issues between competitors with deliberation over a period of months. This may lead to greater reluctance on the part of contractors to make the necessary investments to create the best product at the lowest cost to the government. Rather, the contractors may focus on locating production in states which have powerful Congressional representatives, rather than the states which have the lowest cost or which are otherwise more appropriate for production. If this, indeed, occurs, it could lead to a reduction in innovation, since the focus will have shifted from the quality of the product to the importance of political considerations within Congress. Indeed, it reduces the importance of having a transparent and well-documented government procurement process if Congress can ultimately block the funding for the winning proposal anyway.

7. Conclusion

As of this writing, the outcome of the tanker contract has not been settled. Boeing has lodged a protest, and the GAO is reviewing the case. While the work on the tankers has temporarily been halted, the outcome of Boeing's appeal will likely become clear in the next few months.

The award of the contract to the Northrop/EADS team was significant for several reasons. First, it indicated that the Air Force was anxious to get the best product at the lowest cost. EADS' willingness to innovate was seen in other competitions and in its R&D to create a new boom. Both its innovative tendencies and its flexibility are hallmarks of a successful entrant into a new industry, while Boeing's focus on its pre-existing tanker models and the degree to which they met the specifications stated by the Air Force is indicative of the behavior of a traditional incumbent. Second, the Air Force's willingness to award the contract to a team involving a foreign contractor suggests the recognition that the defense industry has become a global industry that is sufficiently robust to be able to respond to a range of threats to the security of the global community. Third, it indicated the Air Force's willingness to defend its position and describe its criteria emphasizes the transparent and well-documented nature of the process.

The protests against the award, however, have emphasized the fact that the Northrop/EADS team includes a foreign competitor. The discussions and statements of many of the opposing Congressional representatives have focused on the need to prevent American jobs from going overseas, despite the fact that the Northrop/EADS contract would create some jobs in the US, especially in the port of Mobile, AL. This type of argument is often made to protect a declining industry or a failing incumbent against lower-cost, more innovative products made by industries overseas. It encourages the placement of temporary bandages on the problem, rather than an exploration into the heart of why the industry or firm in question is uncompetitive or the development of strategies to make the industry or firm successful. Boeing's own arguments in its official protest, however, have focused more on the differences between the two products and the guidance that it received from the Air Force than on the issue of US jobs going overseas—this latter argument has been made more by Congressional representatives in the affected areas.

In conclusion, the tanker competition embodies many of the key debates across industries in the US economy. Changes in the overall US industrial base, rising fuel prices, the weakness of the dollar, and the range of threats confronting the global community, including the threat posed by terrorism, are important forces in making a procurement decision. Hopefully, the outcome which best serves the American people and the US military will emerge from the dialogue between Boeing, Northrop Grumman/EADS, the Air Force, the GAO, and Congress and will reinforce the move towards more transparent processes, the best product at the lower cost, and the recognition of a more global defense environment.

List of References

Analysts assess damage to Boeing in aftermath of contract loss. (2008, March 3). *Marketwatch*. Retrieved March 18, 2008, from http://www.marketwatch.com

Boeing, Bell team get $204.5M Navy deal. (2008, March 5). *Associated Press*. Retrieved March 18, 2008, from http://biz.yahoo.com

Boeing calls for Air Force review of tanker contract awarded to Northrop. (2008, March 3). *Los Angeles Business.* Retrieved March 18, 2008, from http://losangeles.bizjournals.com

Boeing: Far from satisfied. (2008). *Business Week.* Retrieved March 18, 2008, from http://www.businesweek.com

Boeing protests U.S. Air Force tanker contract award. (2008, March 11). Press Release. Retrieved March 18, 2008, from http://biz.yahoo.com

Boeing requests immediate KC-X tanker briefing. (2008, March 4). Press release from the Boeing Corporation. Retrieved March 18, 2008, from http://biz.yahoo.com

Boeing shares rise on new orders. (2008, March 14). *Associated Press.* Retrieved March 18, 2008, from http://biz.yahoo.com

Boeing study projects that as oil prices climb, 767 tanker most cost efficient. (2008, March 17). *Press Release*. Retrieved March 18, 2008, from http://biz.yahoo.com

Boeing to challenge U.S. Air Force tanker decision. (2008, March 10). *Reuters*. Retrieved March 18, 2008, from http://biz.yahoo.con

Boeing to protest $35B tanker deal. (2008, March 11). *CNNMoney.com*. Retrieved March 18, 2008, from http://CNNMoney.com

Boeing’s trouble with tankers. (2008). *Business Week.* Retrieved March 18, 2008, from http://www.businessweek.com

Boeing wins $77M Air Force contract. (2008, February 25). *Associated Press*. Retrieved March 18, 2008, from http://biz.yahoo.com

Boeing wins $32.8 million Air Force contract. (2008, March 13). *Associated Press.* Retrieved March 18, 2008, from http://biz.yahoo.com

Boeing wins $28.2M Navy contract boost. (2008, March 14). *Associated Press*. Retrieved March 18, 2008, from http://biz.yahoo.com

Carpenter, D. (2008, March 5). Boeing says its tanker proposal superior. *Associated Press.* Retrieved March 18, 2008, from http://biz.yahoo.com

Caterinicchia, D., & Tessler, J. (2008, March 3). Boeing hits bumps, analysts parse path. *Associated Press*. Retrieved March 18, 2008, from http://biz.yahoo.com

Conn. wants answers from Air Force. (2008, March 6). *Associated Press.* Retrieved March 18, 2008, from http://biz.yahoo.com

Crawley, J., McSherry, M., Rigby, B., & Vorman, J. (2008, March 10). Boeing to challenge US Air Force tanker decision. *Reuters*. Retrieved March 18, 2008, from http://biz.yahoo.com

Crown, J., & Epstein, K. (2008, March 11). Boeing files tanker protest. *Business Week*. Retrieved March 18, 2008, from http://www.businesweek.com

Daly, M. (2008, March 3). McCain still considering tanker deal. *Associated Press*. Retrieved March 18, 2008, from http://www.buz.yahoo.com

Drawbaugh, K. (2008, February 29). Congress in turmoil over Air Force tanker decision. *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

Drawbaugh, K. (2008, March 3). U.S. Congress roiled by Air Force tanker decision. *Reuters.* Retrieved March 18, 2008, from http://www.reuters.com

Drawbaugh, K. (2008, March 11). US Lawmaker drafting bill to block tanker funding. *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

EADS confirms tanker refueling test. (2008, March 4). *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

Hepher, T. (2008, March 3). EADS shares soar after big US defense deal. *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

Hepher, T. (2008, March 5). France says EADS deal reflects better US ties. *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

Hepher, T. (2008, March 7). Airbus CEO rejects backlash over US tankers deal. *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

Hepher, T. (2008, March 11). EADS 2007 net loss worse than expected. *Reuters*. Retrieved March 18, 2008, from http://biz.yahoo.com

Hinton, C. (2008, February 22). Air Force set to award $40 bln air tanker contract. *Marketwatch*. Retrieved March 18, 2008, from http://www.marketwatch.com

Hinton, C. (2008, March 10). US Air Force debriefs Northrop Grumman on tanker award. *Marketwatch.* Retrieved March 18, 2008, from http://www.marketwatch.com

Hinton, C. (2008a, March 11). Boeing files protest over lost US tanker deal. *Marketwatch*. Retrieved March 18, 2008, from http://www.marketwatch.com

Hinton, C. (2008b, March 11). Boeing files protest over the Air Force tanker award. *Marketwatch.* Retrieved March 18, 2008, from http://www.marketwatch.com

Hinton, C. (2008, March 12). Boeing might deliver just 45 Dreamliners in 2009. *Marketwatch.* Retrieved March 18, 2008, from http://www.marketwatch.com

Kan. Senate seeks to halt Air Force deal. (2008, March 7). *Associated Press*. Retrieved March 18, 2008, from http://biz.yahoo.com

Lagorce, A. (2008, March 3). Northrop-EADS tanker wins a boon for Europe. *Marketwatch*. Retrieved March 18, 2008, from http://www.marketwatch

Lagorce, A. (2008, March 7). EADS mulls changes to block higher foreign stakes. *Marketwatch*. Retrieved March 18, 2008, from http://www.marketwatch.com

Morgan, D. (2008, February 29). US says little about European role in tanker deal. *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

Northrop group wins $35B Air Force deal. (2008, February 29). *CNNMoney.com*. Retrieved March 18, 2008, from http://www.CNNMoney.com

Northrop Grumman fires back on tanker debate. (2008, March 5). *CNBC.com.* Retrieved March 18, 2008, from http://www.cnbc.com

Palmer, D. (2008, March 6). U.S. fighting Airbus case at WTO despite tanker deal. *Reuters.* Retrieved March 18, 2008, from http://biz.yahoo.com

Pfeifer, S. (2008, March 28). EADS wins 13 billion pound RAF tanker contract. *Financial Times,* p. 21.

Raytheon gets US Air Force contracts. (2008, March 17). *Associated Press.* Retrieved March 18, 2008, from http://biz.yahoo.com

Rigby, B. (2008, March 3). Boeing faces questions after tanker loss. *Reuters*. Retrieved March 18, 2008, from http://www.biz.yahoo.com

Rigby, B. (2008, March 11). Boeing says Air Force tanker award flawed. *Reuters.* Retrieved March 18, 2008, from http://biz.yhoo.com

Shalal-Esa, A. (2008), March 4). Air Force agrees to brief Boeing on tanker loss. *Reuters*. Retrieved March 18, 2008, from http://biz.yahoo.com

Tanker deal: Why Boeing shouldn’t protest. (2008). *Business Week.* Retrieved March 18, 2008, from http://www.businesweek.com

Tessler, J. (2008, February 22). Huge stakes in Air Force tanker deal. *Associated Press*. Retrieved March 18, 2008, from http://biz.yahoo.com

Tessler, J. (2008, March 6). Air Force tanker can’t be all American. *Associated Press.* Retrieved March 18, 2008, from http://biz.yahoo.com

Tessler, J. (2008, March 11). Boeing protests tanker decision to GAO. *Associated Press*. Retrieved March 18, 2008, from http://biz.yahoo.com

Vandore, E. (2008, March 3). EADS shares surge on $35B US Contract. *Associated Press.* Retrieved March 18, 2008, from http://www.biz.yahoo.com

Vandore, E. (2008, March 10). Airbus parent eyes US acquisition. *Associated Press.* Retrieved March 18, 2008, from http://www.biz.yahoo.com

Vorman, J., & Wolf, J. (2008, March 6). Boeing likely to decide on tanker protest this weekend. *Reuters*. Retrieved March 18, 2008, from http://biz.yahoo.com

Wingfield, B. (2008, March 11). Boeing fights back. *Forbes.com*. Retrieved March 18, 2008, from http://www.forbes.com

Witkowski, W. (2008, March 3). GE expects $5 bln for engines in Northrop contract. *Marketwatch*. Retrieved March 18, 2008, from http://www.marketwatch.com

Wolf, J. (2008, February 28). US Air Force Chief extends tanker suspense. *Reuters*. Retrieved March 18, 2008, from http://www.reuters.com

Wolf, J. (2008, February 29). US Air Force to announce tanker award at 5 p.m. *Reuters.* Retrieved March 18, 2008, from http://www.reuters.com

Wolf, J., & Shalal-Esa, A. (2008, March 2). Northrop-Eads beats Boeing to build US tanker. *Reuters*. Retrieved March 18, 2008, from http://biz.yahoo.com

Wolf, J. (2008, March 7). Boeing gets U.S. briefing on tanker decision. *Reuters*. Retrieved March 18, 2008, from http://biz.yahoo.com

Wolf, J. (2008, March 18). Boeing confident of winning back tanker deal. *Reuters*. Retrieved March 18, 2008, from http://reuters..com

The Changing Shape of the Defense Industry and Implications for Defense Acquisitions and Policy[1]

Presenter: Victoria A. Greenfield, PhD (Agricultural and Resource Economics, UC Berkeley), holds the Crowe Chair in the Economics of the Defense Industrial Base at the US Naval Academy, where she leads two upper-level courses: "The Economics of the Defense Industrial Base" and a capstone research seminar. She has published recently on economically targeted terrorism, risk management in military contracting, contractors on the battlefield, technical barriers to trade, defense aerospace globalization, narcotics trafficking, and replacement policies for aging aircraft.

Previously, Greenfield has held the positions of Senior Economist, RAND Corporation; Senior Economist for International Trade and Agriculture, Council of Economic Advisers, Executive Office of the President; Chief International Economist, Bureau of Economic and Business Affairs, US Department of State; and Principal Analyst, Natural Resources and Commerce Division, Congressional Budget Office. Dr. Greenfield has advised senior US policymakers on wide-ranging issues, including China's entry into the WTO, the negotiation of the Kyoto Protocol, the US-Japan civil aviation agreement, and farm policy under NAFTA.

Victoria A. Greenfield
Crowe Chair Professor
Department of Economics
US Naval Academy
589 McNair Road
Annapolis, MD, 21402
410 293 6896 (office)
571 239 8467 (cell)
vag@usna.edu

Author: Ryan R. Brady
Assistant Professor
Department of Economics
US Naval Academy
589 McNair Road
Annapolis, MD, 21402

Abstract

In the mid-1990s, the US defense industry experienced a dramatic wave of consolidation. This paper seeks to establish the statistical facts of defense industry consolidation, including the ways in which it reshaped the industry in the 1990s; the ways in which it may continue to reshape the industry; and the forces that promote or discourage it. It also seeks to consider the implications of consolidation for defense acquisitions and policy. The paper places the events of the 1990s in the broad context of economic and industrial activity spanning almost five decades: 1958-2006. It draws

[1] This paper, prepared for the Naval Postgraduate School's 5th Annual Acquisition Research Symposium, Monterrey, CA, May 13-15, draws on research presented at the 82nd meeting of the Western Economic Association International, June 29-July 3, 2007 (Greenfield, 2007). The authors wish to thank their colleagues for useful comments on this paper and the earlier draft; however, they take full responsibility for any errors or omissions and for the views expressed in this paper, which are theirs alone.

primarily—and in new ways—from a contracting data set known as the DD350 and applies standard economic models and tools. The paper finds that consolidation has had its most pronounced effects at the highest levels of the industry; that the process of consolidation has abated, if not reversed itself, in recent years; and that larger domestic and international economic force have been at least as important as DoD budget decisions and policy in promoting consolidation. The DoD has a significant say in what happens in the defense industry but cannot control it.

Introduction

In the mid-1990s, the US defense industry experienced a dramatic wave of consolidation. This paper seeks to establish the statistical facts of defense industry consolidation, including the ways in which it re-shaped the industry in the 1990s; the ways in which it may continue to re-shape the industry; and the forces that promote or discourage it. It also seeks to consider the implications of consolidation for defense acquisitions and policy. The paper does not isolate the events of the 1990s; rather, it places them in the broad context of economic and industrial activity spanning a period of almost five decades, 1958-2006. It draws primarily—and in new ways—from a contracting data set known colloquially as the DD350 and makes use of standard economic models and tools. Though taking a US perspective, this paper also considers the contributing role of economic globalization.

The issue of defense industry consolidation took on particular importance in the last decade of the 20th century, a period in which the post-Cold War defense budget shrank and US Department of Defense (DoD) suppliers—especially aerospace-defense firms, but also shipbuilding and other defense-related firms—faced the prospect of declining orders, excess capacity, and rising costs. In theory, consolidation could have resulted in a smaller number of leaner and healthier firms, better able to contain or reduce cost through economies of scale and, with deeper financial pockets, better able to bear risk.[1] However, whether the DoD would have benefited from consolidation would have depended, in part, on how the process of consolidation unfolded, how the DoD wrote and executed its contracts, and how much bargaining leverage it maintained.

In 1993, at a dinner now referred to as the “Last Supper,” the DoD asserted its support for industry-wide consolidation, possibly codifying the inevitable—not enough business to go around then or anytime soon—but also suggesting that, in its view, the effects of consolidation would be positive.[2] The DoD backed its assertion by taking a

[1] Consolidation can occur in at least three dimensions: “physical,” involving the combination or elimination of physical assets; “managerial,” involving the combination or elimination of managerial assets; and “corporate,” involving the combination or elimination of corporate entities. Corporate consolidation, which is the main focus of this paper, may or may not be accompanied by physical or managerial consolidation, though some amount of both, especially managerial, is likely. When physical or managerial consolidation occurs, an industry may shed unneeded infrastructure and overhead and reap economies of scales; corporate consolidation may yield financially stronger firms but less competition.

[2] At least implicitly, DoD’s position suggested that some amount of physical and managerial consolidation would accompany corporate consolidation and that the gains from economies of scale and risk-bearing capabilities would dominate any ill-effects of a reduction in competition. For a recent look back at the dinner event, see Augustine (2006).

more pro-active role in anti-trust policy, (i.e., consulting and advising the lead anti-trust agencies: the US Department of Justice and the Federal Trade Commission), and by reimbursing firms for some consolidation-related costs. The DoD is still involved in the anti-trust policy process, but it discontinued the reimbursements in 1998.[3]

Questions remain as to what happened after 1993 and also what happened before. How consolidated has the defense industry become and is it becoming more or less consolidated over time? Questions also remain as to the causes and ultimate effects of consolidation and whether the DoD should—or can—do anything about it.[4]

A chart, sometimes referred to as the “eye chart” provides one perspective on the course of events (see Figure 1). The chart has become a popular fixture in defense community briefings, sometimes serving as a summary statement for all that has happened in the industry in recent decades.[5] It shows corporate consolidation among aerospace-defense firms from the point of dozens in the 1980s to the point of only a handful, literally, 25 years later. But the chart tells only part of the story. For example, it disregards new entrants; the changing composition of defense spending, including the increased importance of spending on war-related and other services; and the role of globalization. Moreover, it alludes to a potential consequence of consolidation (i.e., less competition) but does not speak to such consequences directly.

[3] The reimbursements, sometimes referred to as “payoffs for layoffs,” and most other visible signs of DoD support for consolidation ended in 1998, when the anti-trust agencies with DoD backing, denied the Lockheed-Martin-Northrop-Grumman merger.

[4] For early assessments of the effects of consolidation on excess capacity, cost savings, and other economic parameters, see GAO (1997), GAO (1998), and Gholz & Sapolsky (1999-2000); Hensel (2007) relates consolidation patterns to cost.

[5] Though completely unscientific as a measure of popularity, the author notes that she has seen this chart or charts like it in nearly every defense industry conference she has attended.

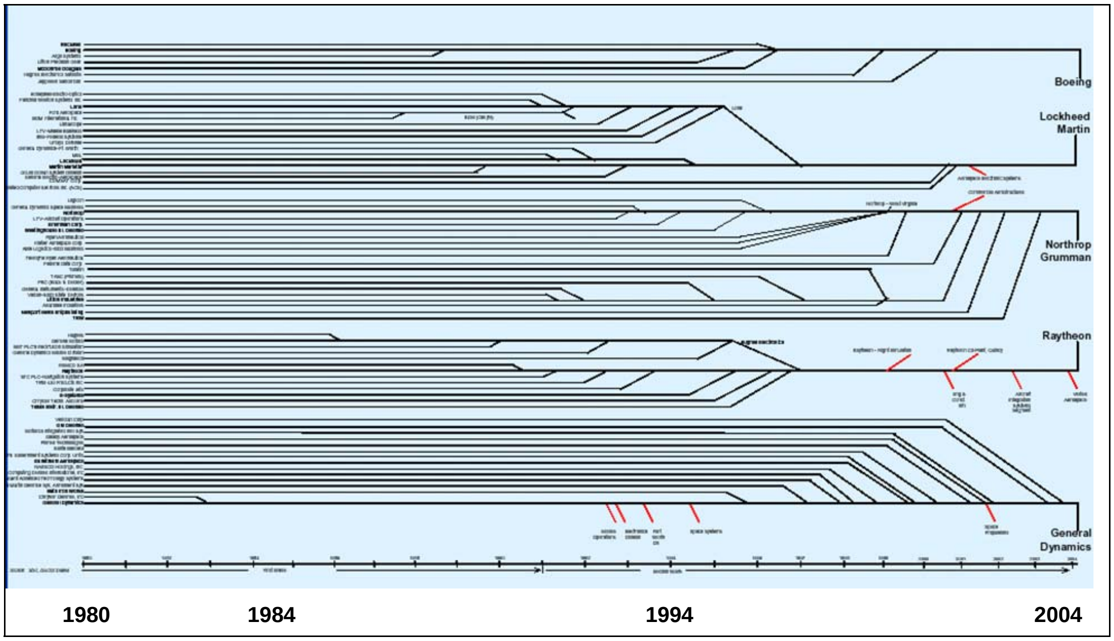


Figure 1. Five Big Players Emerge from Consolidation
(Chao, 2005, p. 13, citing others)

This paper lays a foundation for a more thorough investigation of consolidation, by establishing a broad but tractable definition of "defense industry" and by exploiting a rich but, for these purposes, largely untapped data set.

Defense industry analysts have long debated the meaning of "defense industry," offering wide-ranging, but plausible definitions that generate very different lists of defense or defense-related firms.[6] Chu and Waxman (1998) review several possible definitions and conclude that "the approach we choose must reflect a reasoned choice about the attributes which concern us most" (pp. 35-39). On that basis, this paper takes a DoD-centric approach and defines the industry as those firms that supply goods and services to the DoD or, more generally, those firms that could supply goods and services to the DoD either now or in the future.

This paper draws data from several governmental sources, most notably, a DoD contracting database, commonly known as the DD350, so named for a form completed for each DoD contract action over a certain dollar threshold; an annual DoD publication reporting the awards to the top 100 contractors; various DoD budget documents, including the annual National Defense Budget Estimates; the US Department of Labor, Bureau of Labor Statistics, for estimates of labor productivity; and the US Department of

[6] For example, Dunn (1995, pp. 402-403) suggests classifying defense firms on the basis of their products: lethal large or small weapons systems; non-lethal but strategic products (vehicles and fuel); and other products consumed by the military (food and clothing). He also suggests constructing a matrix of dependencies, one that compares the dependency of the firm on defense-related contracts and the importance of the firm to the military. The greater the mutual dependence, the more readily a firm can be considered a defense firm. One such matrix plots the share of each firm's revenue coming from defense contracts against the share of defense contracts going to each firm (Chu & Waxman, 1998, p. 37).

Commerce, Bureau of Economic Analysis (BEA) and the US Department of Commerce, Bureau of the Census (Census Bureau) for other economic parameters. It also uses data from two non-governmental sources: FactSet Mergerstat, LLC (hereafter, Mergerstat), a firm that specializes in tracking the value and number of mergers and acquisitions, and the Aerospace Industries Association (AIA). The DD350 has known limitations both in terms of coverage and reliability, but it is the best available means of identifying "firms that supply goods and services to DoD," calculating their market shares, and evaluating the extent to which they compete for DoD contracts (see Dixon, Baldwin, Ausink & Campbell, 2005, for a discussion of some of the shortcomings of the DD350).

Trends in Consolidation

Figure 2 shows the number of mergers and acquisitions (M&As) among aerospace-defense firms from 1992 to 2007 and in the overall economy from 1962 to 2007.[7,8] For purposes of scaling, the total value of the economy-wide M&As was an estimated $1,124 billion in 2007, in year-2000 dollars, and about $175 billion in 1968—the first year for which value data are available, also in year-2000 dollars.

[7] Mergerstat defines the aerospace-defense firms as "Aerospace, Aircraft & Defense," consisting of SICs 3721-3728; 3761-3769; and 3795. The data include M&As that involve a US firm as either a buyer or seller. Mergerstat provided the aerospace-defense data for 1992-2006 to the author on February 13, 2007; the economy-wide data and 2007 aerospace-defense data are available online at, https://www.mergerstat.com/newsite/freereport_log.asp .

[8] Mergerstat reports on the number of M&As in each year and their total value, based on the equity prices offered; it assigns M&As to the years in which they are announced. The data on the quantity and value of economy-wide M&As reach back to 1962 and 1968, respectively; the data on the quantity and value of aerospace-defense M&As reach back to 1992.

Mergerstat does not report any values that could reveal proprietary information and, in the smaller aerospace-defense sample, the gaps matter (just over half the 523 aerospace-defense deals reported for the 1992-2006 period had no information on value). The quantity and value series track closely for the overall economy but diverge for the aerospace-defense firms. For this reason, Figure 2 reports only the number of aerospace-defense and economy-wide M&As. For the aerospace-defense firms, the correlation between the value and quantity series was only 0.144 in the 1992-2007 period; for the overall economy, the correlation between the two series was 0.844 in the 1992-2007 period and 0.872 in the 1968-2007 period. The correlations were calculated using year-2000 dollars to eliminate the effects of inflation.

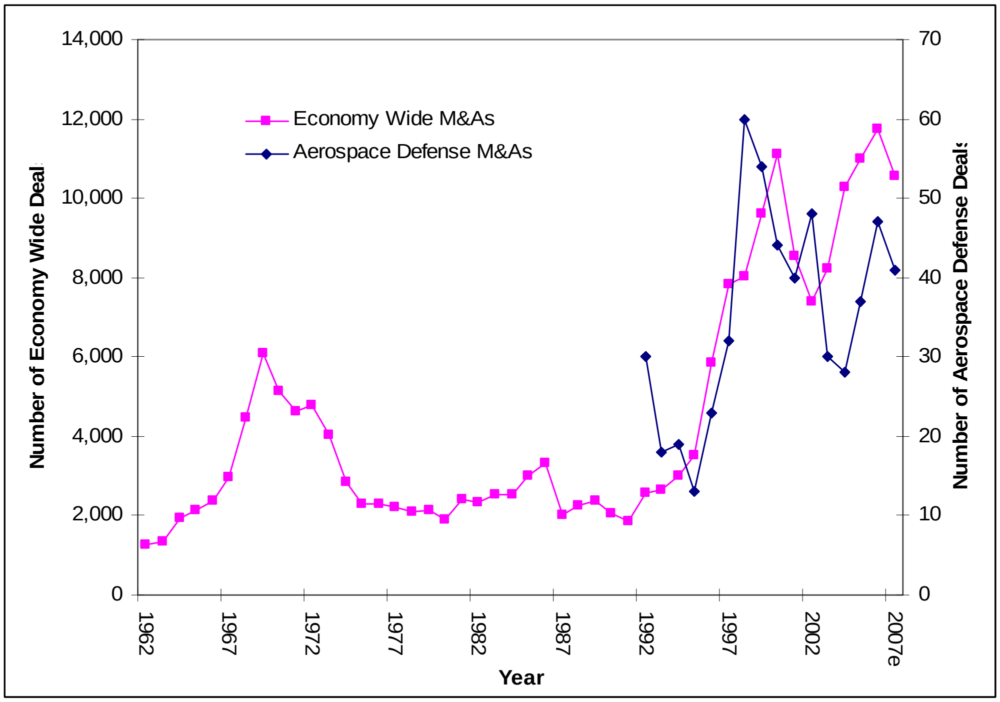


Figure 2. M&As Economy Wide and in Aerospace Defense
(Based on data from FactSet Mergerstat, LLC (a) and (b))

Note that the increases in M&A activity among aerospace-defense firms in the 1990s and 2000s were not unlike those in the overall economy. The later discussion of possible reasons for consolidation addresses this point in statistical detail.

The M&A data in Figure 2 account for the corporate activity that, as a practical matter, would give rise to consolidation, but they say little about the ways in which it unfolds and reshapes an industry (i.e., the defense industry). A simple tally, or even a dollar count, cannot provide this insight; however, the DD350 data and figures from DoD's annual top 100 reports can fill in some of the detail.

The actual DD350 form collects information on DoD contracting actions, which is posted in large text files, by fiscal year, on a central website; the data are currently available from 1966 to 2006.[9] The reporting threshold was $10,000 for actions occurring prior to 1982; $25,000 for actions occurring from 1983-2004; and $2,500 for actions occurring from 2005 to the present. The recent threshold reduction has had a noteworthy effect on the coverage of the data set. In 2004, the data set included filings for just under 700,000 actions; in 2005, it included well over 1.3 million.

Historically, the DD350 has collected information on the timing of the action (month, day, and year), the contractor's name and location, the dollar value of the action, the type of contract, and as many as 70-80 other variables relating to the award process, product, and nature of the business entity, (e.g., small or minority owned). Starting in

[9] Information is available at the following website: http://siadapp.dmdc.osd.mil/procurement/Procurement.html.

1984, the DoD also began assigning an “Ultimate Parent Code” to each contractor, making it possible to track how much business the DoD was doing with each overarching corporate entity in any given fiscal year.[10] On that basis, it is possible to calculate each entity’s annual share of the DoD market, defined here as the dollar value of all DD350 actions (“awards”), and to produce lists of top DoD contractors over time.[11] The DoD has been publishing annual lists of top 100 contractors and their market shares since the 1950s, initially only in hard copy and later only electronically, but the secondary-source data do not contain the full range of information available in the DD350 text files nor can they be manipulated readily.[12] To the extent possible, this analysis draws from the primary data found in the DD350 text files.

Drawing from traditional concepts of both industry and aggregate concentration, this section constructs 4-firm, 8-firm, 20-firm, 50-firm, and 100-firm concentration ratios for the defense industry (i.e., the firms that supply goods and services to the DoD market by fiscal year, for 1958-2006). The concept of “industry concentration” applies insomuch as the analysis focuses on a particular industry; the concept of “aggregate concentration” applies insomuch as it groups firms that, albeit serving a common consumer (DoD), do not necessarily produce like goods or services.[13] The concentration ratio is the market share of each group of firms, in this case the top 4, top 8, top 20, top 50, and top 100, calculated as the group’s share of the net dollar value of all DoD contract awards.[14] The concentration ratio serves as a proxy for consolidation: if, for

[10] Note: The code can—and does—change from year to year so that the data for each year must be handled separately before it can be combined with others in a single time series.

[11] The DD350 reports on actions with prime contractors, thus it does not capture an entity’s indirect market share, via its subcontracts with other contractors. A company that is a prime on a large number of contracts may also be a subcontractor on many others and vice versa.

[12] See DoD, SIAD (2008a), “100 Companies Receiving the Largest Dollar Volume of Prime Contract Awards,” for fiscal years 1996-2006, at http://siadapp.dmdc.osd.mil/procurement/historical_reports/statistics/procstat.html and available in hard copy through the Defense Technical Information Center for 1958-1997.

[13] For a non-exhaustive glimpse into the uses and drawbacks of these types of measures, see Bain (1951), Curry & George (1983), Kwoka (1981), O’Neill (1996), and White (2002).

[14] Why choose 4-, 8-, 20-, 50-, and 100-firm concentration ratios? The choice of the “K-Firm” ratio is inherently arbitrary (Curry and George, 1983, p. 207). The 4-firm and 8-firm ratios are widely accepted in industry studies, so much so, that “AmosWEB”—a.k.a. “The Encyclonomic WEBpedia”—describes them as the “analytical standards in the study of the structure of the industry.” For a more academically compelling justification of the 4-, 8-, 20-, and 50-firm concentration ratios, the US Department of Commerce, Bureau of the Census provides them in the “Economic Census” and other reports that address industry concentration. The calculation of the 100-firm concentration ratio provides a direct bridge to DoD’s top-100 reports and is a common measure in aggregate studies (Curry & George, 1983, p. 207). Lastly, Kwoka (1981) provides strong justification for considering more than one concentration ratio.

example, the concentration ratio is rising for a particular group of firms, then, for purposes of this analysis, that group is becoming more consolidated.[15,16]

The results show a striking increase in the 4- and 8-firm concentration ratios over the 1990s (see Figure 3). In 1990, the top 4 firms accounted for about 18.5% of the DoD contract awards, just above the 17.7% average for the 1960s, ’70s, and ’80s. By 1996, they accounted for 23.6% of the market, breaking the previous “record” of 23.2% in 1958, and by 1999 they accounted for about 28.1%. The ratio was relatively stable through 2003, decreasing in 2004 and 2005, and it rose slightly in 2006, but not to the most recent peak, which was 28.8% in 2002. The calculations for 2005 and 2006 may overstate the decline from 2002 because of the change in the DD350 reporting threshold; however, even after accounting for the change, the concentration ratio likely remained below the 28.8% peak.

Speculatively, the post-2003 decline in concentration ratios may reflect the effects of the Iraqi War. The DoD may be drawing from different firms at different market levels for different types of products (e.g., it may be requiring more sustainment services and ordering fewer new weapon systems). The rise of Halliburton to the top 8 in 2005—it ranked 6th—provides circumstantial evidence.

[15] Concentration ratios have also been used as proxies for competition (O’Neill, 1996); however, this paper considers concentration and the relationship between concentration and competition separately.

[16] The ratios also make partial use of Dunn’s matrix (see footnote 7; Dunn, 1995, pp. 402-403), by establishing a rough measure of DoD’s dependence on suppliers at different market levels.

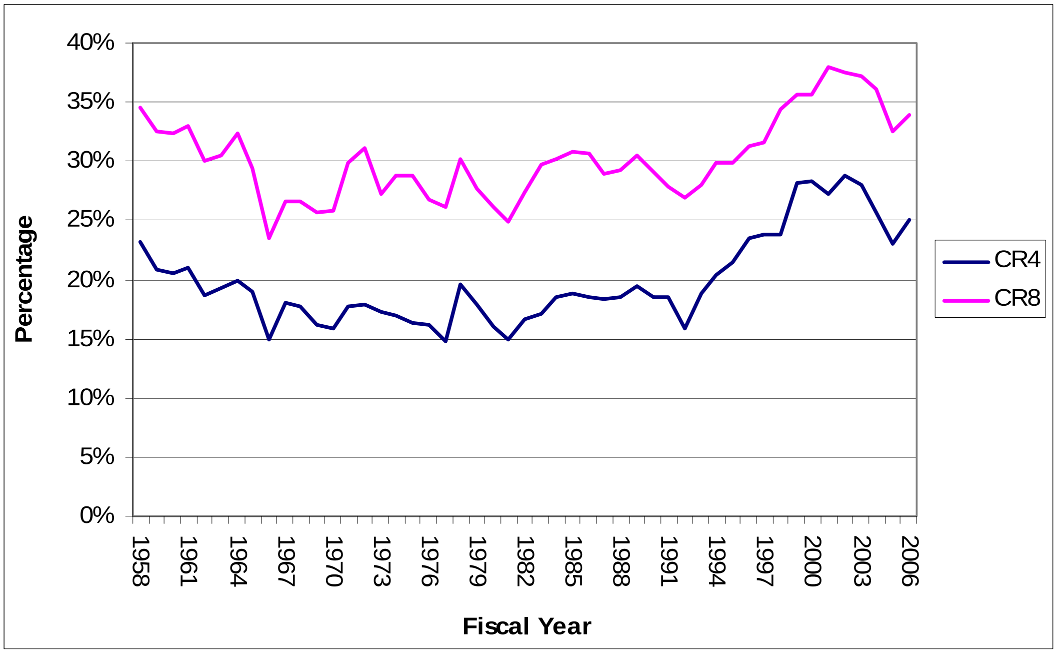


Figure 3. 4- and 8-Firm Defense Industry Concentration Ratios
(Based on data from DoD DD350 (DoD, SIAD, 2008a; 2008b, from 1958-2006))

The 20-, 50-, and 100-firm defense industry concentration ratios did not rise along with the 4- and 8-firm ratios; the 50- and 100-firm ratios declined over much of the 1990s, just as they did through much of the preceding decades and did not begin to rise until the end of the 1990s (see Figure 4). The 50- and 100-firm ratios have also dropped off since 2003, with a slight resurgence in 2006.

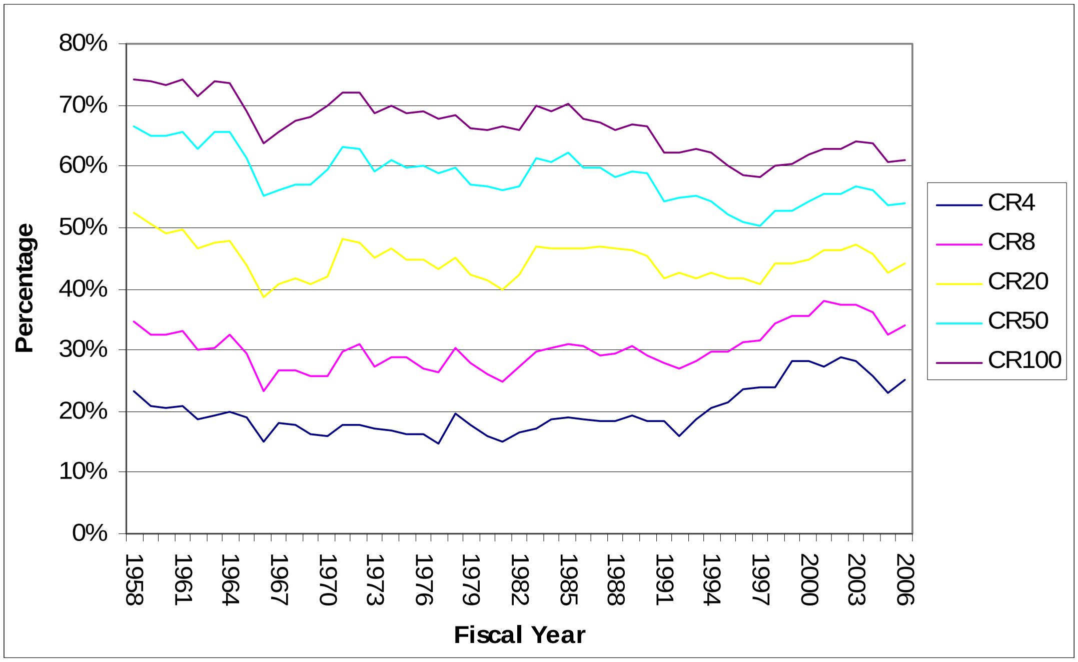


Figure 4. 4-, 8-, 20-, 50-, and 100-Firm Concentration Ratios
(Based on data from DoD DD350 (DoD, SIAD, 2008a; 2008b, from 1958-2006))

A more formal correlation analysis clarifies the relationships among concentration ratios at different market levels. The relationships between the 4- and 8-firm concentration ratios and the 50- and 100-firm concentrations are the strongest. The 4-firm and 8-firm concentration ratios are positively and significantly correlated at 0.926; the 50-firm and 100-firm concentration ratios also are positively and significantly correlated at 0.977. To the extent that the 4- and 8-firm concentration ratios are correlated with the 50- and 100-firm concentration ratios, however weakly, they are negatively correlated. The results pivot around the 20-firm concentration ratio. The highest levels of the industry do not appear to be "behaving" like the next highest levels.

Two alternative data presentations shed additional light on the time-path of industry consolidation and on differences across and within market levels.

The first presentation replaces the concentration ratios of the top 4, top 8, top 20, etc., which are aggregate categories, with marginal market-level breakouts, i.e., the top 1-4, top 5-8, top 9-20, top 21-50, and top 51-100 (see Figure 5).[17] The marginal breakouts highlight substantial differences across and within the market levels. While the "paths" of the top 4 and top 8 firms look nearly identical in Figures 3 and 4, the paths for the breakouts of the top 1-4 and top 5-8 firms in Figure 5 are quite dissimilar. Whereas the top-most firms became increasingly concentrated—a handful of major players became even more major—at lower but-still-quite-high levels, the industry became more diffuse or remained largely as it was in the 1980s and prior decades. For

[17] According to Curry & George (1983, p. 207), R.A. Miller introduced the concept of the marginal concentration ratio in a paper, "Marginal Concentration Ratios and Industrial Profit Rates: Some Empirical Results," published in the *Southern Economic Journal* in October of 1967.

the top 5-50 firms, a downward slide in concentration began in the mid-to-late-1980s, when the DoD budget began its decline (the budget peaked in real terms in 1985). The 51st to 100th firms have experienced less change, particularly in recent decades.[18]

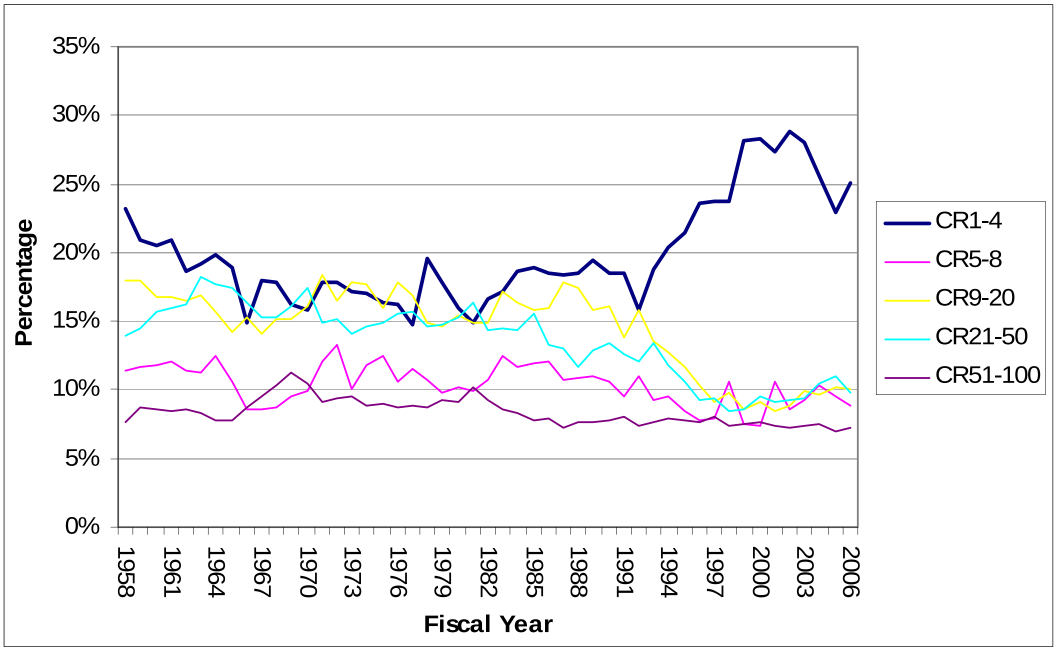


Figure 5. 1-4-, 5-8-, 9-20-, 21-50-, and 51-100-Firm Concentration Ratios
(Based on data from DoD DD350 (DoD, SIAD, 2008a; 2008b, from 1958-2006))

The second presentation compares the market shares of each of the top 50 firms at 5-year intervals (1989, 1994, 1999, and 2004) to their counterparts in 1984 (see Figure 6).[19] The x-axis in Figure 6 shows the rank of the firm from 1 to 50 (based on its share of the DoD market, measured in the dollar value of DoD contract awards), and the y-axis shows the difference between its market share and the share of the equally-ranked firm in 1984. For example, in 1989 the first-ranked firm's market share was almost 1 percentage point higher than that of the first-ranked firm in 1984 (in this case, the same firm, but not necessarily so); in 1994 the first-ranked firm's market share was

[18] Though tempting to equate rank with size to draw conclusions for large, medium, and small firms, rank and size are not equivalent. Firms of similar rank may be quite different in size and firms of less-than-top-most rank may also be quite large. The top-100 firms are the top of literally tens of thousands of firms that serve the DoD market, to rank 50th or 51st is not trivial. In 2006, the 50th firm, Dell, did about $636 million in business with DoD; the 51st firm, American Body Armor, did another $635 million.

[19] The authors initially conducted this analysis using data from 1984, the first year for which the DD350 text files provide ultimate parent company identifiers and are in the process of incorporating data from earlier years. However, a comparison of the top-50 firms' ranks and market shares in the mid-to-late 1980s and around each of the interval years suggests that 1984 provides a reasonable anchor and that the 5-year intervals are representative of overall trends in the industry. The 4-firm concentration ratio for 1984 is also close to the historical average for the decades preceding the 1990s.

2.1 percentage points higher than that of the first-ranked firm in 1984; in 1999 the first-ranked firm's market share was 4.3 percentage points higher than that of the first-ranked firm in 1984; and in 2004 the first-ranked firm's market share was 3.2 percentage points higher than that of the first-ranked firm in 1984.

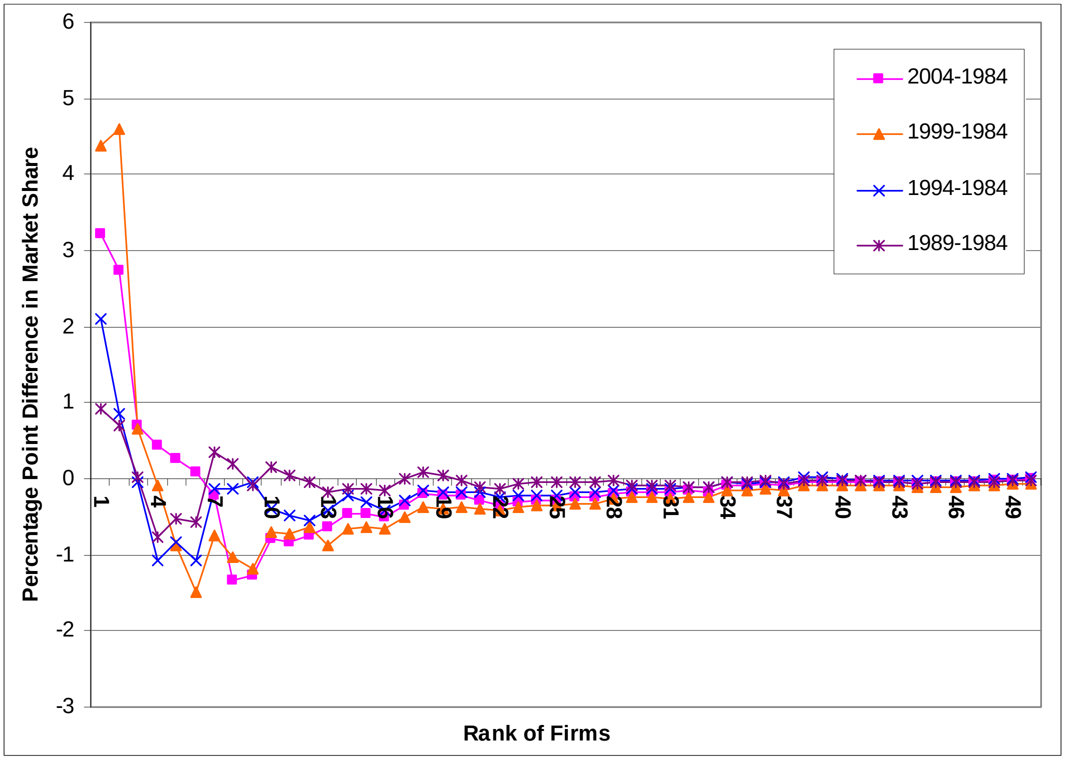


Figure 6. Changes in Market Share by Firm Rank
(Based on data from DoD DD350 (DoD, SIAD, 2008b, from 1984-2004))

The differences in market shares grew over the 1990s but then subsided in the 2000s. The differences in shares are most pronounced among the top-20 firms, especially the top 10, and all but vanish by the 50th firm. Lastly, the differences for the top-5 to top-50 firms are often negative, not positive, which implies that, for all but the very top-most firms, the market became more diffuse, not less. The change in distribution might be consistent with consolidation among the DoD's very top-most firms and a "hollowing out" of upper-to-mid-ranking firms, perhaps by absorption, but also with equal or heightened competition among all but the very top-most firms. For firms ranked outside the top 50, the market shares in 1989-2004 look like "business as usual."

Possible Explanations

Why did the defense industry's top-most firms consolidate in the 1990s, and why has consolidation shown sign of abating more recently? It is tempting to view the numbers and say "it's all about the defense budget" or "it's all about DoOD policy," but the seemingly obvious could be wrong and potentially misleading. Why the red flag?[20] First, US defense budgets, including procurement budgets, have seen substantial downturns in the past without inspiring dramatic consolidation (see Figure 7), and second, the defense industry does not operate in a vacuum. Yes, demand for defense products contracted when the Cold War ended declined, possibly so-much-so that it could no longer sustain the then-larger number of major contractors, and, yes, DOD reportedly announced its support for consolidation during the Last Supper of 1993, but the rest of the US economy was also experiencing a dramatic wave of corporate activity, particularly M&As (Emmons & Perry, 1997). Other factors may also have been drivers.

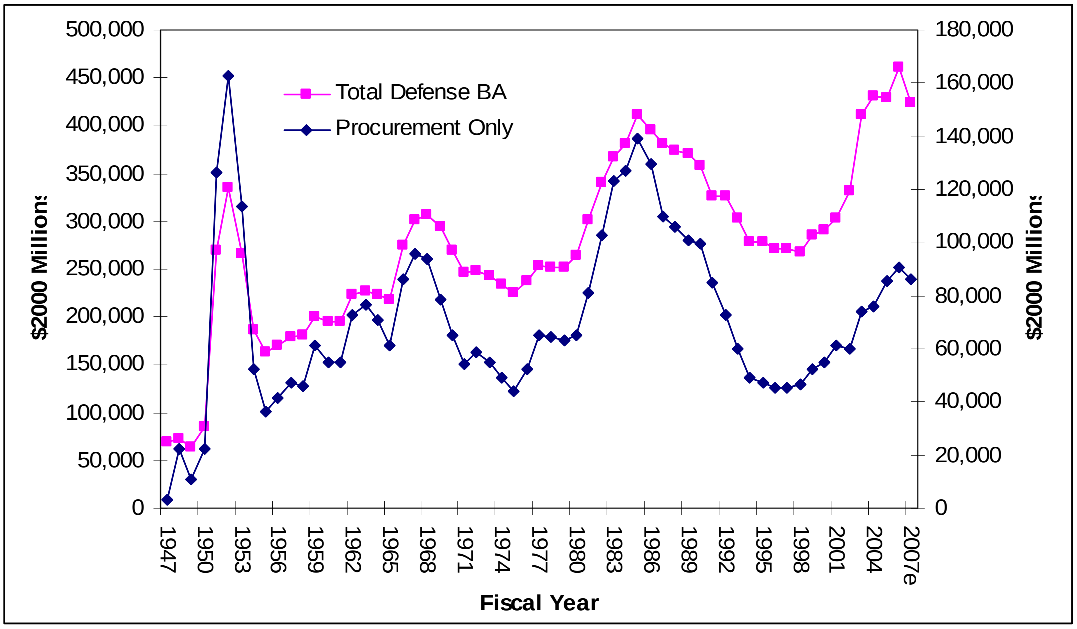


Figure 7. DoD Budget Authority in 2000 Dollars
(Based on data from the DoD National Defense Budget Estimates for the FY 2007 and 2008 Budgets and Department of Commerce, Bureau of Economic Analysis)

A simple empirical model considers the possibility of multiple contributing factors by regressing the 4-firm concentration ratio for 1958-2006 on combinations of both economy-wide and DoD-specific variables, including real gross domestic product (GDP), the number of M&As in all industries, real DoD budget authority (BA), and a dummy variable for DoD policy, to capture the effects of the Last Supper and subsequent DoD policy actions, including cost reimbursements.

[20] Other analysts have also questioned popular wisdom. Hensel (2007) suggests that that economy-wide exuberance in the 1990s might have been responsible for the wave of M&As in the defense industry and assesses the correlations between aerospace-defense M&As, economy-wide M&As, and DoD outlays in the 1992-2004 period. Flamm (1998, pp. 45-46) notes that the aircraft, aircraft engines, and munitions industries became more consolidated in the 1980s, during a period in which spending “soared.” Note: This analysis does not report an increase in overall defense-industry concentration during that period; rather, it suggests stable or declining market shares.

Several statistical challenges arose in the analysis, including missing data, i.e., the absence of economy-wide M&A data for 1958-1961; evidence of autocorrelation and collinearity; and the presence of a non-specific upward trend over most of the period of analysis. Initial data runs indicated positive first-order serial correlation at a significance level of 0.01, suggesting the importance of including a lagged industry concentration variable among the independent variables. Initial data runs also indicated likely collinearity among the independent variables, especially among GDP, economy-wide M&As, lagged industry concentration, and a proposed trend term.[21] Ultimately, a model that included economy-wide M&As and excluded GDP offered the strongest statistical results but necessitated a 1962 start date (n = 45).[22]

$Y = B_0 + B_1X_1 + B_2X_2 + B_3X_3 + B_4X_4 + B_5X_5$

Where:

Y = 4-firm concentration ratio (in decimal terms, e.g., 0.18, 0.25, etc.) (CR4F)

X_1 = Lagged 4-firm concentration ratio (one period lag) (CR4F-L)

X_2 = Lagged real DoD BA (in $2000 billions) (BA-L)

X_3 = DoD policy (0, 1 dummy) (POL)

X_4 = Number of economy-wide M&As (MA)

X_5 = Trend term (Linear, 1…N) (TR)

Table 1 shows the results of six model runs. The first considers only one independent variable, X_1, the lagged 4-firm concentration ration (CR4F-L). The second run adds the trend term, X_5. The third run adds X_2, lagged real DoD BA (BA-L). The lag allows firms to adjust to new information about the budget and for changes in BA to begin to transform themselves into changes in actual spending and contracting actions.[23] The fourth run adds X_3, the policy variable (POL), defined as "1" for 1993-1997 (arguably, the period in which DoD most actively promoted consolidation) and "0" for all

[21] We model the concentration ratio with a deterministic time trend, implying the process is trend-stationary. Dickey-Fuller and Phillips-Perron tests of stationarity support this modeling choice. Specifically, a process is trend-stationary if the de-trended series is stationary (see Hamilton, 1994, p. 435). We also tested for a cointegrating relationship between the concentration ratio and the BA series or the MA series. The Johansen procedure failed to find evidence of cointegration for either pair of variables.

[22] Starting with either 1958 or 1962 and substituting other economy-wide variables, such as GDP, the Dow Industrial Average or corporate profits, yields similar but statistically weaker results, which tend to conflate the roles of the economic variables and the underlying positive trend.

[23] Eliminating the lag on BA does not have a dramatic effect on the results nor does substituting other lagged or un-lagged measures of defense spending, including total outlays and various measures of procurement, research and development, and operations and maintenance.

other years.[24] The fifth run adds X_4, the number of economy-wide M&As (MA). The 6th and final run removes the policy variable. A priori, one might expect the signs on the X_1 (CR4F-L), X_2 (BA-L), X_3 (POL), X_4 (MA), and X_5 (TR) coefficients to be positive, negative, positive, positive, and positive, respectively.

[24] Some have described the discontinuation of reimbursements and the denial of the Lockheed-Martin-Northrop-Grumman merger in 1998 as a change in DoD policy (e.g., Flamm, 2005), but it may be more accurate to describe it as a continuation of the same policy in the face of a change in circumstance. The DoD had said that it would support consolidation to the point that it made sense, and, in 1998, it may have stopped making sense. Nevertheless, be it a labeled a change of “policy” or “circumstance,” something appears to have changed in the DoD’s behavior in 1998.

Table 1. Regression Results for Concentration Ratio Model

	Intercept	**CR4F-L**	**BA-L**	**POL**	**MA**	**TR**
(1)	**B_0**	**B_1**	**N/A**	**N/A**	**N/A**	**N/A**
Coefficient	0.18	0.913				
(t-stat)	(1.297)	(13.110)				
Test results	R^2(adj.) = 0.795; F = 171.878; DW = 2.082					
(2)	**B_0**	**B_1**	**N/A**	**N/A**	**N/A**	**B_5**
Coefficient	0.034	0.753				0.001
(t-stat)	(2.337)	(8.452)				(2.626)
Test results	R^2(adj.) = 0.820; F = 101.169; DW = 2.060					
(3)	**B_0**	**B_1**	**B_2**	**N/A**	**N/A**	**B_5**
Coefficient	0.059	0.710	-7.976E-5			.001
(t-stat)	(2.693)	(7.685)	(-1.515)			(3.024)
Test results	R^2(adj.) = 0.825; F = 70.291; DW = 2.133					
(4)	**B_0**	**B_1**	**B_2**	**B_3**	**N/A**	**B_5**
Coefficient	0.051	0.736	-6.315E-5	.007		.001
(t-stat)	(2.099)	(7.336)	(-1.086)	(0.694)		(2.165)
Test results	R^2(adj.) = 0.823; F = 52.172; DW = 2.153					
(5)	**B_0**	**B_1**	**B_2**	**B_3**	**B_4**	**B_5**
Coefficient	0.069	0.609	-6.653E-5	0.008	2.628E-6	0.001
(t-stat)	(2.709)	(5.144)	(-1.180)	(0.885)	(1.883)	(1.817)
Test results	R^2(adj.) = 0.834; F = 45.103; DW = 2.109					
(6)	**B_0**	**B_1**	**B_2**	**N/A**	**B_4**	**B_5**
Coefficient	0.077	0.582	-8.682E-5		2.516E-6	0.001
(t-stat)	(3.283)	(5.101)	(-1.690)		(1.815)	(2.737)
Test results	R^2(adj.) = 0.835; F = 56.490; DW = 2.081					

(Based on data from DoD DD350 (DoD, SIAD, 2008a; 2008b, from 1958-2006), DoD National Defense Budget Estimates for the FY 2007 and 2008 Budgets, the Department of Commerce, Bureau of Economic Analysis; and FactSet Mergerstat, LLC (b), using SPSS v16)

The results of the 1st regression indicate that lagged industry concentration accounts for a large share of the variance in industry concentration.[25] Nevertheless, the underlying statistical process may be more than just an autoregressive "Black Box." Focusing on the results of the 6th and final regression:

- The coefficient on the DoD BA variable is statistically significant, at the 0.10 confidence level, and signed as expected (if BA decreases by a billion dollars in one year, the 4-firm concentration ratio increases by about 0.00009 in the next year, indicating a small but non-negligible increase, for a budget that moves in increments of multiple billions of dollars.)
- The coefficient on the economy-wide M&A variable is also statistically significant and signed as expected (all else constant, the increase in economy-wide M&As in 2006, would have been associated with an increase in the 4-firm concentration ratio of about 0.002, also a small but non-negligible increase).
- Lagged industry concentration and economy-wide M&As are significantly correlated, but fortunately this is one of the "happy situations" that Gujarati discusses in his text (2006, p. 377). The variables' collinearity neither eliminates statistical significance nor results in "incorrect" signage.

In the 4th and 5th regressions the coefficient on the DoD policy variable is not statistically significant and, in an unreported regression that focuses solely on policy, by excluding the DoD BA variable, it is not significant either.[26] The budget data may say all that needs to be said about DoD's role in shaping the industry—hard dollars may be more important than proclamations, even when proclamations come with changes in policy processes and the possibility of cost reimbursements.

Given the predictive strength of the lagged defense-industry concentration variable, it is striking that both BA and economy-wide M&As find moderate predictive significance. The jump from "predictor" to "driver" is a long and dangerous one; however, the results suggest a role for the defense budget and the general economy in shaping events. The declining DoD budget may have been an important driver of defense industry consolidation in the 1990s but perhaps not the only driver. Ironically, it may have been a peculiar combination of economy-wide exuberance and defense-wide doldrums that led to the industry's reformation.[27,28]

[25] The standardized coefficient (Beta) for CR4F-L is 0.894 in the first regression; 0.738 in the second regression; 0.696 in the third; 0.721 in the fourth; 0.597 in the fifth; and 0.571 in the sixth.

[26] The unreported regression includes lagged industry concentration, policy, and the trend term.

[27] For insight to the underlying causes of the economy-wide wave, including the roles of deregulation and governance, see Andrade, Mitchell & Stafford (2001) and Holmstrom & Kaplan (2001). Note that past authors have not necessarily found a strong correlation between M&As and industry or aggregate concentration ratios; for example, see Curry & George (1983) and White (2002).

[28] A closer look at the industrial composition of DOD's top firms suggests another possible driver requiring further consideration: an aging or "maturing" industry, like aircraft manufacturing, may have less ability—or need—to sustain a large number of leading firms than a newly developing

Over the past few years, the 4-firm concentration ratio has tracked more closely with the DoD budget, which has been growing,[29] and less closely with economy-wide M&As, which have been largely increasing. It is also tracking less closely with aerospace-defense M&As, which may speak to the rising importance of sustainment services and other war-related purchases in the spending mix.

These results are not just academically interesting, they are also relevant to policy making. The DoD may be the sole purchaser of some or even many defense-related products, thus, highly influential in its purchase habits, but its purchase habits do not solely determine the fate of the industry. The DoD's control over the firms that arm, feed, and clothe it is by no means absolute. Defense firms, even including the firms that are most dependent on the DoD, are also captive to larger market forces: they seek financing in global markets and must satisfy their investors. Need be, they can opt out entirely—in the extreme, by terminating their operations and liquidating their assets—if the opportunity cost of doing business with the DoD becomes too great. With that in mind, the following section briefly considers the possible implications of consolidation.

Implications for Defense Acquisitions

Building on a long-standing tradition, this section relies heavily on standard economic models and tools to consider the implications of defense industry consolidation for DoD acquisitions (for some early examples, see Adams & Adams, 1972; Bohi, 1973; Peck & Scherer, 1962; Stigler & Friedland, 1971; and Weidenbaum, 1968).[30,31]

Concerns about consolidation stem in large part from those about deviations from competition (see Ackerman, Giovachino, Tighe & Trunkey, 1995): are firms in the defense industry consolidating that are behaving less competitively and, if they are, will their behavior have a pronounced effect on the price, quantity, quality, or timeliness of deliveries in the near term or on productivity and innovation in the longer term?

Consolidation and Competition

A simple market model and a preliminary assessment of data on the extent of competition in contract awards shed light on static concerns about price and quantity.

industry. Whether the industry's relative maturity would have contributed positively, negatively, or not at all to the corporate activity of the 1990s remains an open empirical question.

[29] In the context of a negative relationship, to "track" really means to move in opposition.

[30] Peck & Scherer (1962) have an entire chapter addressing "The Nonmarket Character of the Weapons Acquisition Process"; nevertheless, they conduct an economic analysis of the acquisition process, relying almost exclusively on standard economic—and market—principles.

[31] Occasionally, skeptics argue that the defense industry is unique; that it has no real market for its products; and by extension, that standard economic models and tools have little practical use. This skepticism may reflect a misunderstanding both of the nature of the "standard economic models and tools" and of how "uniqueness" enters into economic analysis. See Johnson (1958) for a related study of the possible uniqueness of agricultural markets. The issue is not whether an industry is unique but whether standard models and tools can identify and incorporate its uniqueness.

The Simple Market Model

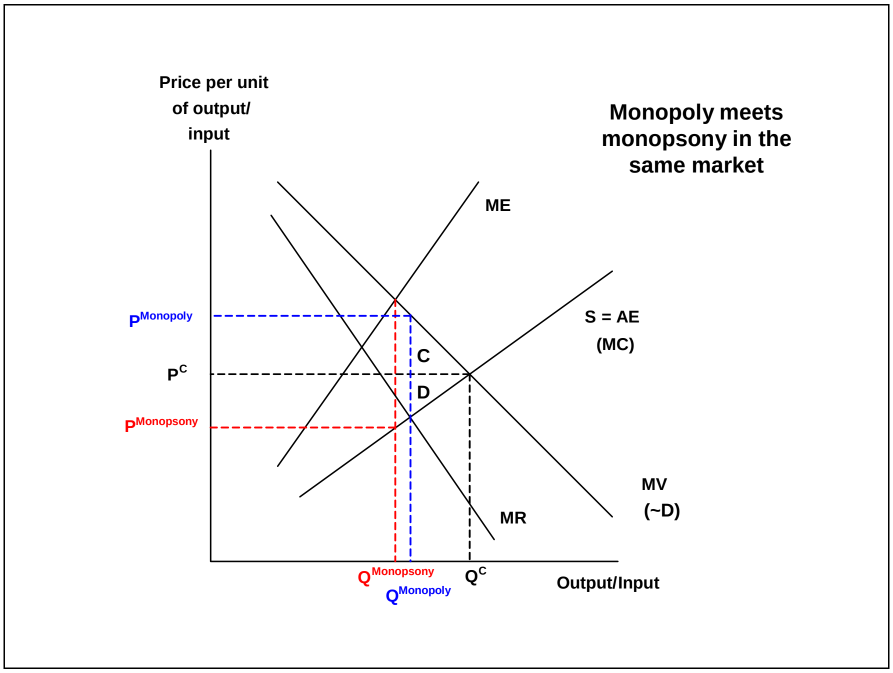
The “simple market model” is a model of bilateral monopoly. In the bilateral monopoly a single buyer, a monopsonist, meets a single seller, a monopolist (Pindyck & Rubinfeld, 2001, pp. 358-359 and pp. 525-527). Prices and quantities are indeterminate but bound by the standard monopsonist and monopolist outcomes. The model is highly stylized. In most instances, the sole-buyer-sole-seller assumption is an exaggeration; however, the model provides a basis for considering the effects of convergence toward unity on either side of the market.[32] To the extent that defense industry consolidation reduces competition among producers of particular defense-related products, particularly those with a single primary purchaser (i.e., the DoD), the model provides insight.

Figure 8 depicts the marriage of the monopsony and monopoly. In a case that concerns a pre-existing monopsony or quasi-monopsony, such as the DoD, and the possible convergence of the supply side of the market to monopoly, the relevant comparison is not that of the bilateral monopoly and a perfectly competitive market but of the bilateral monopoly and a stand-alone monopsony. Nevertheless, in comparing the two imperfect markets, perfect competition provides a useful reference point.

When a monopsony faces competitive suppliers, it faces the entire market supply curve because it is the only purchaser in the market;[33] when it purchases more of the product in question, price rises; and when it purchases less, price falls. It recognizes the effect of its purchases on price and so it purchases less than it would in a perfectly competitive market. Although the equilibrium price is lower than in a perfectly competitive market, the quantity is also lower.

[32] For a small set of defense-related products, such as nuclear-powered aircraft carriers, pure monopsony and pure monopoly provide a reasonably accurate characterization.

[33] The monopsonist does not have a true demand curve; rather, it has a marginal valuation (MV) curve. This is analogous to the case of the monopolist, supply, and marginal cost.



Notes: S = supply; AE = average expense associated with each unit of purchase or input, which is also the industry's supply (S) curve; MC = marginal cost; ME = marginal expense associated with each unit of purchase or input; MV = marginal value to the purchaser, which is shown as D = demand for the perfectly competitive market; MR = marginal revenue; Pc = the equilibrium price in the perfectly competitive market; Pmonopsony = the equilibrium price in the monopsony-only market; Pmonopoly = the equilibrium price in the monopoly-only market; Qc = the equilibrium quantity in the perfectly competitive market; Qmonopsony = the equilibrium quantity in the monopsony-only market; Qmonopoly = the equilibrium quantity in the monopoly-only market. Absent the monopoly, the monopsonist would equate ME and MV and sets price and quantity along AE, the industry's supply curve. Absent the monopsony, the monopolist would equate MC and MR and set price and quantity along D, the purchasers' demand curve. The deadweight loss associated with the monopoly, absent the monopsony, would be the area C+D, which in this depiction would be somewhat less than that which would be associated with the pure monopsony.

Figure 8. Monopoly Meets Monopsony

(Based on data from Pindyck & Rubinfeld, 2001, pp. 347, 358-359, and 525-527)

In the bilateral monopoly, price can settle anywhere at or between Pmonopsony and Pmonopoly, including at the perfectly competitive level; however, if the firm has any negotiating leverage vis-à-vis DoD, the bilateral monopoly price will be higher than the monopsony-only price and possibly closer to the competitive price. Quantity will settle somewhere below the perfectly competitive level because neither the monopsonist nor the monopolist has an incentive to exceed it; however, depending on the relative slopes of the various curves, it may settle above or below the monopsony-only quantity. As depicted in Figure 8, the introduction of monopoly to the erstwhile-monopsony-only market could result in a higher price and higher quantity, but with different relative slopes; it could also result in higher prices and lower quantities.

The new-found market power of the monopolist firm would likely result in a transfer of some surplus from the purchaser (i.e., the DoD) to the firm, which, though not a "good thing" from the DoD's perspective in the short term, could have positive consequences if it implies the increased viability of the industry in the long term. The net effect on society-at-large is ambiguous and context specific. Society may actually gain surplus from convergence to monopoly.

The model becomes more complicated with economies of scale, the promise of which provided some or much impetus for the DoD's promotion of consolidation in aerospace, shipbuilding, and elsewhere. In such instances, convergence to monopoly may result in lower prices and greater efficiency than would occur in the more traditionally-depicted bilateral monopoly. If firms consolidate, so that each produces at greater volume, they can move down their respective cost curves and capture those economies. By definition, unit costs drop with increases in output—the cost of production may be $2.2 billion per ship if two firms produce 4 ships each and only $2 billion per ship if one firm produces all 8 ships—so that one firm can produce the same total output at lower total cost than two or more.[34] The monopsonist bids-up prices with each additional purchase, but additional purchases also drive down cost. To the extent that the monopsonist can extract some of the benefits of the cost reductions, its purchases may have a less pronounced price-raising effect than in the simple case.

The foregoing model suggests that, *if* an increase in industry concentration results in less effective competition among firms, then the prices of affected defense-related products may rise and quantities may fall or rise. Ultimately, the price and quantity outcomes will depend partly on the DoD's negotiating abilities; it will have lost some but not all of its relative market power. The outcomes will also depend on whether firms in the defense industry experience economies of scale. Moreover, and perhaps surprisingly, society at large may find itself with more economic surplus than under conditions of pure monopsony, even in the absence of economies of scale.

Preliminary Empirical Assessment

The question remains, however, as to whether the market has become less competitive. Does an increase in concentration among top-ranked firms necessarily imply a loss of effective competition? One way to approach this question is to identify changes in the DoD contracting practices. Are fewer contracts awarded competitively now than in the past, particularly to industry leaders? Admittedly, a decrease in competitive awards may not be causally related to an increase in defense industry concentration, even if contemporaneous and highly correlated;[35] however a clear pattern in the data (or lack thereof) may lend credence to (or refute) a relationship.

This section evaluates the frequency and distribution of one DD350 variable, "Extent Competed" (A = Competed; B = Not Available for Competition; C = Follow on to Competed Action; and D = Not Competed), among top 100 firms; future analyses will also consider four additional variables: "Number of Offerors Solicited," "Number of Offers Received," "Solicitation Procedures," and "Authority for Other than Full and Open Competition." The DD350 provides reasonably complete coverage for "Extent Competed," with consistent documentation and formatting, starting in 1989, but provides only limited coverage for the other variables.

[34] From Pindyck & Rubinfeld (2001, p. 227), "output can be doubled for less than a doubling of price." A natural monopoly is said to experience "strong" economies of scale.

[35] Just as in the case of concentration itself, many "drivers" may explain the difference in practices.

The evaluation begins with data in 5-year increments, comparing the shares of contract dollars awarded competitively, either directly (“Competed”) or indirectly (“Follow on to Competed Action”) to firms operating at different market levels, e.g., top 4, top 8,… top 100, and overall, in 1989, 1994, 1999, and 2004. It then provides additional detail for the years between 1989 and 1994. It also considers the extent of competition in each year in relation to industry concentration. The results are preliminary for at least two reasons: first, because of coding issues in the underlying data and, second, because of mechanical difficulties sorting the data that could lead to error. Future rounds of analysis will address both concerns.

Table 2 shows the share of DoD contract dollars competitively awarded, either directly or indirectly (hereafter the “competitive share”), by year and market level.

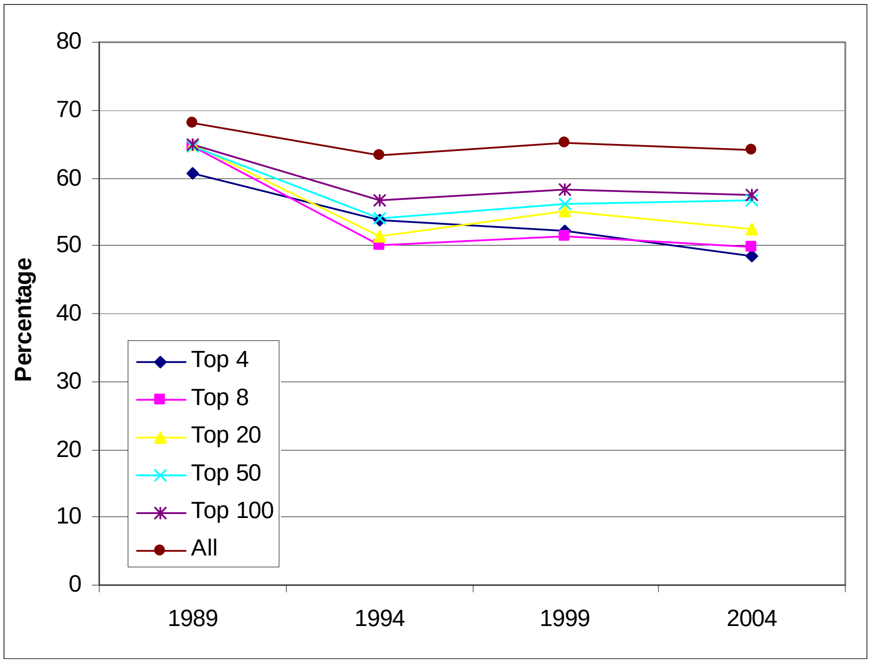
Table 2. The Extent of Competition over 5-Year Intervals
(Based on data from DD350 (DoD, SIAD, 2008b, from 1989-2004))

	Top 4	**Top 8**	**Top 20**	**Top 50**	**Top 100**	**All**
1989	60.72	64.71	65.00	64.51	64.86	68.06
1994	53.83	49.96	51.50	53.92	56.56	63.22
1999	52.20	51.40	55.16	56.04	58.21	65.13
2004	48.40	49.86	52.41	56.80	57.39	64.16
	Top 1-4	**Top 5-8**	**Top 9-20**	**Top 21-50**	**Top 51-100**	**101+**
1989	60.72	71.74	65.56	62.72	67.61	74.50
1994	53.83	41.59	55.14	62.63	74.56	74.24
1999	52.20	48.42	70.53	60.55	74.05	75.67
2004	48.40	53.46	61.97	75.99	61.84	75.99

Note: This table shows the percentages of DoD contract dollars competitively awarded, either directly or indirectly, in aggregate categories and for marginal market-level breakouts.

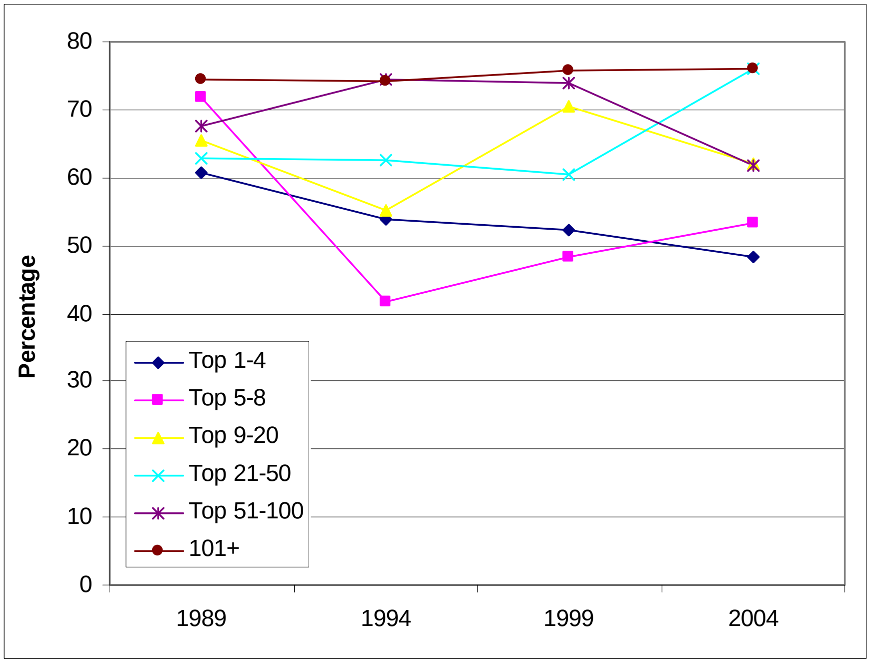
The results for the incremental years (1989, 1994, 1999, and 2004) indicate a consistent downward trend in the extent of direct and indirect competition among the DoD’s top-most suppliers (i.e., the aggregate top 4); however, as previously, the results across and within market levels are mixed. Each of the top 8, top 20, etc., aggregate categories experienced a decline in competitive awards from 1989 to 1994 and a partial rebound in 1999. The differences among the breakouts are more pronounced: the top 21-50 became more competitive over time and the top 51-100, after experiencing a substantial increase in competitive awards in the 1990s, saw a large drop in 2004.

Figures 9 and 10, which show the competitive shares for the top 4, 8, 20, etc., aggregate categories (top panel) and for the top 4, top 5-8, top 9-10 etc., marginal market-level breakouts (bottom panel), starkly illustrate the relative and absolute “mixedness” of the results across and within market levels.



Note: This figure shows the percentages of DoD contract dollars competitively awarded, either directly or indirectly, in aggregate market-level categories.

Figure 9. A Comparison of Competitiveness by Aggregate Market Levels
(Based on data from DD350 (DoD, SIAD, 2008b, from 1980-2004))



Note: This figure shows the percentages of DoD contract dollars competitively awarded, either directly or indirectly, for marginal market-level breakouts.

Figure 10. A Comparison of Competitiveness for Marginal Market Levels
(Based on data from DD350 (DoD, SIAD, 2008b, from 1989-2004)).

Even within the top 4, the results are mixed. Though not show in Table 2 or Figures 9 and 10, the first-ranked firm's competitive share was 49, 39, 65, and 55% in each of 1989, 1994, 1999, and 2004, respectively.[36] Contrary to expectation, the firm's competitive share peaked just as its market share was cresting (its share of the DoD market was over 10% in 1999 and reached over 11% in 2000).

Recalling the earlier finding of an increase in diffusion among the top 5-8, 9-20, and 21-50 firms over the 1990s, the data speak inconsistently to the extent of competition at those levels. The top 5-8 firms saw a dramatic decline in the share of competitive awards from 1989 to 1994 and then a substantial rebound from 1994 to 1999 and again from 1999 to 2004; the top 9-20 saw large fluctuations over the same period; the top 21-50 saw an overall increase; and the top 51-100 saw an increase in the 1990s and a decline in the 2000s. Moreover, whereas the market-share results previously suggested "business as usual" for the top 51-100 firms throughout the 1990s and into the 2000s; these competitive-share results seem to shift "business as usual" to the next market level (i.e., to the 101st firm) and those ranking behind it.

The one nearly consistent pattern to be found in the data is the dramatic change—mostly a drop—in the extent of competition between 1989 and 1994. The change raises questions about the usefulness of 1989 as an anchor point (was it an outlier?) and about the nature of the transition (if 1989 was not an outlier, was the transition smooth or sudden, e.g., did it occur in close proximity to the Last Supper). Table 3 includes data for 1989 and 1994 and the years between them.

[36] Note: The first-ranked firm in 1989 and 1994 was McDonnell Douglas; the first-ranked firm in 1999 and 2004 was, as it is today, Lockheed Martin.

Table 3. Competitive Contract Awards in a Transition Period?

(Based on data from DD350 (DoD, SIAD, 2008b, from 1989-2004))

	Top 4	**Top 8**	**Top 20**	**Top 50**	**Top 100**	**All**
1989	60.72	64.71	65.00	64.51	64.86	68.06
1990	57.77	61.09	60.06	61.38	62.27	65.85
1991	53.75	59.22	59.24	59.75	61.35	64.88
1992	64.65	57.72	57.34	58.59	59.67	63.45
1993	52.67	50.89	51.51	54.92	56.62	61.78
1994	53.83	49.96	51.50	53.92	56.56	63.22
	Top 1-4	**Top 5-8**	**Top 9-20**	**Top 21-50**	**Top 51-100**	**101+**
1989	60.72	71.74	65.56	62.72	67.61	74.50
1990	57.77	66.85	58.22	65.80	69.05	72.95
1991	53.75	69.92	59.26	61.45	72.12	70.73
1992	64.65	47.72	56.67	63.01	67.64	69.65
1993	52.67	47.27	52.79	65.49	68.79	70.48
1994	53.83	41.59	55.14	62.63	74.56	74.24

Note: As above, this table shows the percentages of DoD contract dollars competitively awarded, either directly or indirectly, in aggregate categories and for marginal market-level breakouts.

Looking more closely at the data for 1989 through 1994, the results for the top 4, 8, 20 etc., aggregate categories suggest a reasonably smooth glide path from 1989 to 1994 for all but the top 4, but not for the marginal market-level breakouts.

The data for the 1989-1994 transition suggest that 1989 was not necessarily an outlier and that competition dropped off at some market levels in the years between 1989 and 1994, but that the path from 1989 to 1994 was neither entirely smooth nor tied to a particular event, be it the Last Supper or any other policy action.

Lastly, and perhaps somewhat more concretely, an analysis of the correlation between competition and concentration—via an analysis of the correlation between the shares of DoD contract dollars competitively awarded and the market shares of DoD's suppliers—suggests a negative relationship between competition and concentration for the top 4 firms, in aggregate, but not necessarily for the others (see Table 4).

Table 4. Correlations between Competition and Concentration
(Based on data from DD350 (DoD, SIAD, 2008b, from 1989-2004))

Top 4	**Top 8**	**Top 20**	**Top 50**	**Top 100**	
-0.5599	-0.3211	0.5675	0.8261	0.7834	
Top 1-4	**Top 5-8**	**Top 9-20**	**Top 21-50**	**Top 51-100**	**101+**
-0.5599	0.4420	-0.4021	0.0027	0.5513	0.0890

Note: This table shows the correlations between the competitive shares and market shares at each market level, both for aggregate categories and marginal market-level breakouts.

With data for a small number of years it is difficult, if not impossible, to pull a clear and compelling story from the statistical results.[37] Even within the top 4, the results are mixed. Recalling the noteworthy increase in the first-ranked firm's competitive share after 1994, the correlation between competition and concentration appears to be moderately positive at 0.5034.

On balance and notwithstanding the unexpected results for the first-ranked firm, it seems that competition declined among the top-most firms in the 1990s, but at other market levels the trends in competition are less clear. The lack of clarity may stem from a lack of data, as the analysis currently "misses" the years 1995-1998, 2000-2003, and 2005-2006, or from the underlying methodology. The analysis works with concentration ratios for firms in a buyer-defined (DoD) industry that do not necessarily produce like goods and services; thus, the ratios provide, at best, imperfect proxies for concentration among producers of particular product lines. To the extent that purchases of particular types of items drive the data on competition, the analysis may fail to identify the relevant relationships between concentration and competition.

Consolidation, Productivity, and Innovation

This section offers a brief discussion of the issues surrounding consolidation, productivity, and innovation, while addressing some initial observations on trends in consolidation and productivity in the aircraft industry, as a rough proxy for the larger industry.[38] Future research should look more closely at both the theoretical and empirical relationship between consolidation and productivity and between consolidation and innovation.

In considering productivity and innovation, the case can be argued for or against negative relationships with consolidation. The case "for" may be more familiar: a reduction in competition (e.g., via an increase in consolidation) would result in reductions in productivity (or its growth) and innovation because firms have less need to improve either. The case "against" may be less familiar: post-consolidation firms have more profits to allocate to productivity and innovation-enhancing activities and have incentives to do so because they fear the threat of future competition. In the extreme, firms may

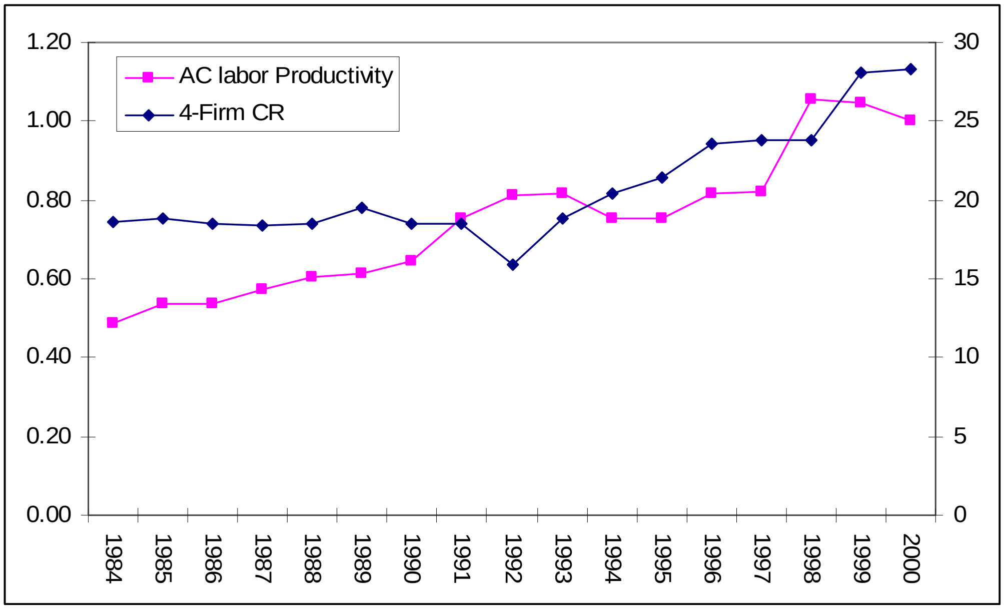
[37] The correlation calculations include data for 1995, bringing the total number of observations to nine (1989-1995, 1999, and 2004) for each market level.

[38] BLS publishes annual data on labor productivity in the aircraft industry but does not publish such data for a more broadly defined defense-aerospace or defense industry.

undertake productivity and innovation-enhancing activities, such as R&D, until they have fully expended the “excess” profits they are seeking to preserve.

A cursory look at the data on aircraft labor productivity (available from 1972-2000) and defense industry concentration[39] (available from 1984-2006) shows an overall upward trend in both (in the period of overlap from 1984-2000), with some opposing movement. A closer look at correlations between aircraft labor productivity and defense industry concentration, controlling for steady increases in manufacturing productivity, lends some statistical weight, albeit far from conclusive, to a negative relationship between aircraft labor productivity and industry concentration.

Figure 11 traces aircraft labor productivity and the 4-firm defense industry concentration ratio from 1984 through 2000.



Note: The aircraft labor productivity variable is an index (2000 = 1).

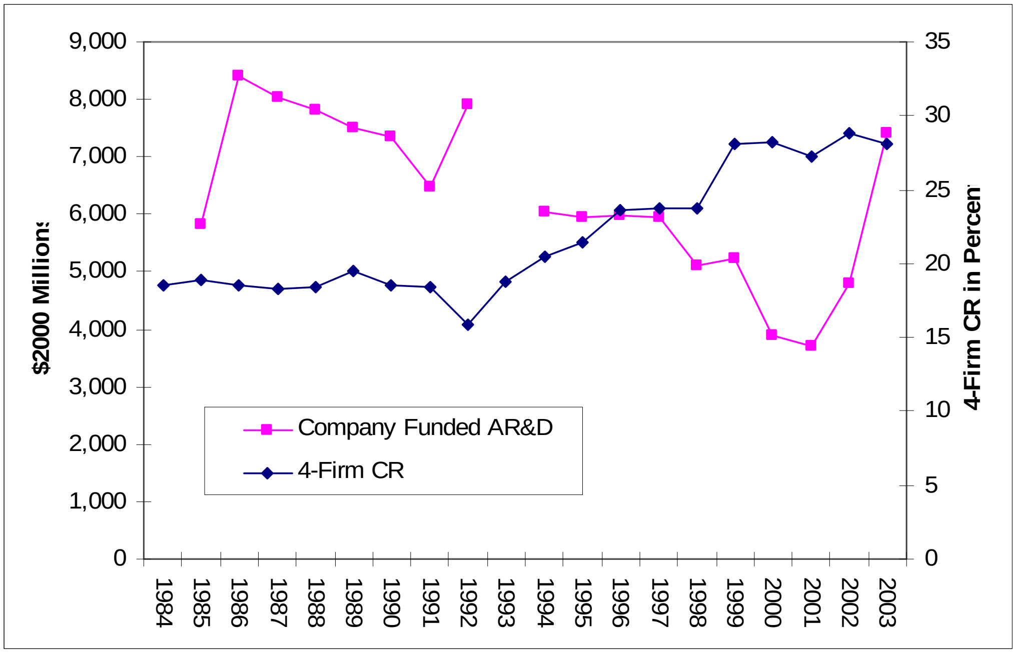
Figure 11. Aircraft Labor Productivity and Defense Industry Concentration
(Based on data from US Department of Labor (2005) and DD350 (DoD, SIAD, 2008b, from 1984-2004))

The correlations among the three variables are all strongly positive, ranging from 0.753, between aircraft labor productivity and the 4-firm concentration ratio, and 0.947

[39] This analysis uses the 4-firm concentration ratio as a proxy for aircraft industry concentration, both of which have been trending upward; however, the defense industry—writ large, in terms of DoD contract awards—is considerably less concentrated, in absolute terms, than the aircraft industry. The Census Bureau (see http://www.census.gov/epcd/www/concentration.html) typically publishes estimates of aircraft industry concentration every five years. In the US market, the four largest aircraft manufacturers accounted for over 80% of shipments in the late 1990s and just less than 60% in the late 1950s; the top eight accounted for well over 90% in the late 1990s.

between aircraft labor productivity and manufacturing productivity. But, the partial correlation between aircraft labor productivity and the 4-firm concentration ratio, after controlling for the contemporaneous rise in manufacturing labor productivity, is actually negative, -0.572, and moderately significant

An even more cursory look at the data on R&D expenditures (more precisely, company-funded applied research and development or AR&D) in the aircraft industry and defense industry concentration shows a stronger negative correlation, i.e., about -0.73 (see Figure 12); however, a careful analysis of the relationship between innovation and concentration would require a much closer examination of all R&D expenditures and other measures of innovation, such as patent awards, in relation to industry concentration, holding a host of other industry and economic variables constant.



Note: AR&D = applied research and development; CR = concentration ratio.

Figure 12. Aircraft AR&D Expenditures and Defense Industry Concentration
(Based on data for company funded applied research and development from the Aerospace Industries Association (2005, p. 105) and DoD DD350 (DoD, SIAD, 2008b, from 1984-2003), using a BEA deflator)

The data on concentration and competition do not paint a clear picture of negative or positive relationships across and among market levels; however, for the top 4 aggregate category, they generally seem to move in opposition. To the extent that this can be taken as a sign of less competition at that level, the relationships between productivity and 4-firm concentration and innovation and 4-firm concentration, may reflect underlying relationships between both variables and competition. At risk of overreaching with inadequate data, the correlations hint at the possibility of negative relationships and provide no immediate evidence to the contrary.

Conclusions and Suggestions for Future Research

In some respects, the "eye chart" is right. It speaks a simple, plausible, and possibly self-evident truth. The DoD's top-most suppliers (i.e., the top 4), in aggregate, have increased their market share and become less competitive since the mid-to-late 1980s. However, the "truth" of the chart is less evident beyond the top 4 and even within the top 4. For example, the first-ranked firm may hold a larger share of the market now than in the 1980s; indeed, it has held a larger share of the market over the past 10 years than at any time since 1958, but it also faces competition, either directly or indirectly, for a larger share of its DoD dollars. Underlying differences in the first-ranked firms in each era may help explain the result. In the 1980s, McDonnell Douglas may have produced a narrower range of products that were not competed openly; whereas, Lockheed Martin may now produce a wider range of products that are competed openly.

Together, the concentration ratios and competitive shares provide an aggregate view of an industry defined in terms of a single buyer: the DoD. In so doing, they enable consideration of the forest; however, it may be helpful to begin to view the trees. Future research will fill in the missing years in the analysis of competition (i.e., 1995-1998, 2000-2003, and 2005-2006) and complete the assessment of competition and concentration, looking more closely at product lines.

The eye chart also fails to fully capture the dynamism of the market. For example, it fails to recognize new entrants and the role of globalization, which are sometimes one and the same. (For example, BAE, a British-founded firm has dramatically increased its presence in the US market through a combination of direct investment in the United States and trade. It broke the top 8 in 2005.) Another new-entrant issue involves the changing composition of defense spending and the increasing prominence of service providers, especially post 9-11 and in Iraq. Noting that the top 51-100 firms and the 101st and beyond have largely experienced "business as usual," the data suggest both continuity and churn: some firms may be absorbed—as may be their fondest desire—and others may enter the market anew. As above, consideration of industry concentration in terms of particular product lines may be fruitful.

To the extent that the top-most level of the market has become less competitive, the simple, static bilateral monopoly model suggests that DoD may be paying more for some products, but the quantity implications are ambiguous. Moreover, if DoD's suppliers experience economies of scale, then the DoD may actually be paying less for some goods and services post-consolidation. Perhaps surprisingly, society-at-large may be economically better off with consolidation even without economies of scale.

However, dynamic concerns about productivity and innovation suggest a need for a closer examination. A cursory look at data on labor productivity and R&D in the aircraft industry suggests the possibility of a negative relationship vis-à-vis consolidation. A rough-and-ready look at correlations among pertinent variables is no basis for conclusions, but it is a reason to look more closely in the future.

Lastly, the statistical analysis of industry concentration and contributing factors suggests, on the one hand, the shared authority of DoD and the larger economy in determining the fate of the industry and, on the other hand, the relative lack of importance of policy proclamations, even when they come with changes in policy processes and potential cost reimbursements. DoD may have a significant say in what happens in the defense industry, but it cannot control the market altogether.

Looking to the future, the implications of the statistical assessment are striking. At present, the defense industry—as it is reflected in the 4-firm concentration ratio—is more closely tracking defense spending (in the sense that spending is rising and the 4-firm concentration ratio is still substantially below peak) than the larger economy; however, were spending to decline (e.g., at the end of the war in Iraq) the conditions might be right for an increase in consolidation.

List of References

Ackerman, G.H., Giovachino, M.J., Tighe, C.E., & Trunkey, R.D. (1995). *Performing industrial base analysis, volume I: Methodology* (CRM 95-112). Alexandria, VA: Center for Naval Analysis.

Adams, W., & Adams, W.J. (1972). The military-industrial complex: A market structure analysis. *American Economic Review*, *62*(1/2), 279-287.

Aerospace Industries Association. (2005). *Aerospace: Facts & figures, 2005-2006*. Arlington, VA: Aerospace Industries Association of America, Inc.

Andrade, G., Mitchell, M., & Stafford, E. (2001). New evidence and perspectives on mergers. *The Journal of Economic Perspectives*, *15*(2), 103-120.

Augustine, N.R. (2006, June 26). The last supper, revisited: Meeting ignited inevitable consolidation. *DefenseNews.com*. Retrieved March 31, 2008, from http://integrator.hanscom.af.mil/2006/June/06292006/06292006-13.htm

Bain, J.S. (1951). Relation of profit rate to industry concentration: American manufacturing, 1936-1940. *The Quarterly Journal of Economics*, *65*(3), 293-324.

Bohi, D.R. (1973). Profit performance in the defense industry. *The Journal of Political Economy*, *81*(3), 721-728.

Chao, P.A. (2005, March 2). *Structure and dynamics of the defense industry*. Prepared for the MIT Security Studies Program Seminar, Cambridge, MA.

Chu, D.S.C., & Waxman, M.C. (1998). Shaping the structure of the American defense industry. In G.I. Susman & S. O'Keefe (Eds.), *The defense industry in the post-Cold-War era: Corporate strategies and public policy perspectives* (pp. 35-44). London: Pergamon Press.

“Concentration Ratios.” (2007). AmosWEB, the Encyclonomic WEBpedia. Retrieved June 12, 2007, from http://amosweb.com

Curry, B., & George, D. (1983). Industry concentration: A survey. *The Journal of Industrial Economics*, *31*(3), 203-255.

Dixon, L., Shirley, C., Baldwin, L.H., Ausink, J.A., & Campbell N.F. (2005). *An assessment of Air Force data on contract expenditures* (MG-274-AF). Santa Monica, CA: RAND.

DoD, OASD (Comptroller), Directorate for Information Operations (and Control). (n.d.). *100 companies receiving the largest dollar volume of prime contract awards, fiscal years 1956-1976*. Washington, DC: Author.

DoD, Statistical Information Analysis Division (SIAD). (2008a). *100 companies receiving the largest dollar volume of prime contract awards, fiscal years 1996-2006*. Retrieved March 31, 2008, from http://siadapp.dmdc.osd.mil/procurement/historical_reports/statistics/procstat.html

DoD, Statistical Information Analysis Division (SIAD). (2008b). *DD350 data, forms, and layouts, fiscal years 1984-2006*. Retrieved March 31, 2008, from http://siadapp.dmdc.osd.mil/procurement/Procurement.html

DoD, Washington Headquarters Services, Directorate for Information Operations and Reports. (n.d.). *100 companies receiving the largest dollar volume of prime contract awards, fiscal years 1977-1997.* Washington, DC: Author.

Dunn, J.P. (1995). The defense industrial base. In K. Hartley & T. Sandler (Eds.), *Handbook of defense economics.* (Vol. I). Amsterdam, the Netherlands: Elsevier Press.

Emmons, G., & Perry, N.O. (1997). Blockbuster deals, M&As make a comeback. *HBS Bulletin Online*. Retrieved March 31, 2008, from http://www.alumni.hbs.edu/bulletin/1997/june/deals.html

FactSet Mergerstat, LLC (a), a global provider of mergers and acquisitions information. *FactSet Mergerstat volume report for aerospace, aircraft and defense* (SICs 3721-3728; 3761-3769; and 3795) *for 1992-2006*. Provided through personal communication with Mergerstat on February 13, 2007.

FactSet Mergerstat, LLC (b), a global provider of mergers and acquisitions information. *Mergerstat free reports: M&A activity, US and US cross-border transactions*. Economy-wide data for 1962-2006 retrieved February 22, 2007, from http://www.mergerstat1.com/newsite/freereport_log.asp; data for 2007, including aerospace, aircraft, and defense data, retrieved January 24, 2008, from https://www.mergerstat.com/newsite/free_report.asp

Flamm, K. (1998). US defense industry consolidation in the 1990s. In G.I. Susman & S. O'Keefe (Eds.), *The defense industry in the post-Cold-War era: Corporate strategies and public policy perspectives* (pp. 45-69). London: Pergamon Press.

Flamm, K. (2005). Post Cold War policy and the US defense industrial base. *The Bridge*, *35*(1). Retrieved March 31, 2008, from http://www.nae.edu/NAE/bridgecom.nsf/

GAO. (1997). *Defense industry restructuring: Cost and savings issues* (GAO/T-NSIAD-97-141). Washington, DC: Author.

GAO. (1998). *Defense industry: Restructuring costs paid, savings realized, and means to ensure benefits* (GAO/NSIAD-99-22). Washington, DC: Author.

Gholz, E., & Sapolsky, H.M. (1999-2000). Restructuring the defense industry. *International Security*, *24*(3), 5-51.

Greenfield, V. (2007). *Wither goest the US defense industrial base?: The changing shape of the defense industry and implications for defense acquisitions*. Prepared for the 82nd meeting of the Western Economic Association International.

Gujarati, D.N. (2006). *Essentials of econometric* (3rd ed.). New York: McGraw-Hill-Irwin.

Hamilton, J.D. (1994). *Time Series Analysis*. Princeton, NJ: Princeton University Press.

Hensel, N. (2007). An empirical analysis of the patterns in defense industry consolidation and their subsequent impact. In *Proceedings of the fourth annual acquisition research symposium*. Monterey, CA: Naval Post Graduate School.

Holmstrom, B., & Kaplan, S.N. (2001). Corporate governance and merger activity in the United States: Making sense of the 1980s and 1990s. *The Journal of Economic Perspectives*, *15*(2), 121-144.

Johnson, D.G. (1958, October). Government and agriculture: Is agriculture a special case? *The Journal of Law and Economics, 1*(1), 122-136.

Kwoka, J.E., Jr. (1981). Does the choice of concentration ratio really matter? *The Journal of Industrial Economics*, *29*(4), 445-453.

O'Neill, P.B. (1996). The trend of aggregate concentration in the United States: Problems of scope and measurement. *American Journal of Economics and Sociology*, *55*(2), 197-211.

OUSD (Comptroller). (2006, March; 2007, March). *National Defense Budget estimates for the FY 2007 and 2008 budgets.* Retrieved 2007 and 2008 and available as of March 31, 2008, from http://www.defenselink.mil/comptroller/defbudget/fy2009/index.html

Peck, M.J., & Scherer, F.M. (1962). *The weapon acquisition process: an economic analysis*. Boston, MA: Division of Research, Graduate School of Business Administration, Harvard University.

Pindyck, R.S., & Rubinfeld, D.L. (2001). *Microeconomics* (5th ed.). New Jersey: Pearson-Prentice Hall.

Stigler, G.J., & Friedland, C. (1971). Profits of defense contractors. *American Economic Review*, *61*(4), 692-694.

US Department of Commerce, Bureau of the Census. (n.d.). *Economic census* and various other statistical reports that address industry concentration. Retrieved in 2007 and 2008 and available as of March 31, 2008, from http://www.census.gov/epcd/www/concentration.html

US Department of Commerce, Bureau of Economic Analysis. (n.d.). GDP deflators and other economic statistics. Retrieved in 2007 and 2008 and available as of March 31, 2008, from http://www.bea.gov/national/nipaweb/Index.asp

US Department of Labor, Bureau of Labor Statistics (2005). Industry productivity indexes, by SIC, for aircraft and parts, aircraft, etc., in txt file, IPRSICDATA, dated December 19, 2005, and retrieved January 31, 2007, from http://www.bls.gov/lpc/home.htm

Weidenbaum, M.L. (1968). Arms and the American economy: A domestic convergence hypothesis. *American Economic Review*, *58*(2), 428-437.

White, L.J. (2002). Trends in aggregate concentration in the United States. *The Journal of Economic Perspectives*, *16*(4), 137-160.

THIS PAGE INTENTIONALLY LEFT BLANK

Panel 4 - Spiral Development in Major Defense Acquisition Programs

Wednesday, May 14, 2008	**Panel 4 - Spiral Development in Major Defense Acquisition**
11:15 a.m. – 12:45 p.m.	**Chair:** **Rear Admiral Michael S. Frick, USN**, Vice Commander, Naval Sea Systems Command **Discussant:** **Mr. T. Craig Smith**, Head, Engagement Systems Department, NSWC Dahlgren Division **Papers:** ***Using Spiral Development to Reduce Acquisition Cycle-times*** **Dr. William Lucyshyn**, Director of Research and Senior Research Scholar, Center for Public Policy and Private Enterprise, University of Maryland ***Modeling the Integration of Open Systems and Evolutionary Acquisition in DoD Programs*** **Dr. David Ford**, Associate Professor, Texas A&M University and **John Dillard**, Senior Lecturer, Naval Postgraduate School

Chair: Rear Admiral Michael S. Frick is originally from Fort Meade, MD. He enlisted in the Naval Service in August 1971 under the Nuclear Power Program and received his commission in June 1977. He holds a Bachelor of Science degree from the United States Naval Academy and a Master of Science degree from the United States Naval Postgraduate School.

Upon commissioning, Frick served his initial tour of duty as the Main Propulsion Assistant and then Weapons Officer on USS *Carpenter* (DD 825) until February 1981. He was then assigned as Officer-in-Charge of a Mobile Training Team for transfer of Ex-USS *Carpenter* (DD 825) to the Turkish Navy. He reported to the Naval Postgraduate School in May 1981 majoring in Anti-Submarine Warfare Systems Technology. Graduating in September 1983, he reported to USS *Henry B. Wilson* (DDG 7) as Engineer Officer until August 1986, and then to the USS *William H. Standley* (CG 32) as Engineer Officer until February 1988.

His next assignment was Deputy Program Manager for the Surface Ship Torpedo Defense Program at Naval Ocean Systems Center, San Diego, CA. In June 1990, he reported to USS *Leftwich* (DD 984) as Executive Officer, where he participated in Operations Desert Shield and Desert Storm. Frick's next assignment in January 1992 was Chief Staff Officer for Commander, Destroyer Squadron 31. He served on the staff of the Chief of Naval Operations in Washington, DC, from July 1993 to August 1995—first as Head, Surface ASW Weapons section, and then as Financial Coordinator for the Theater Air Defense Division. He then commanded USS *Curts* (FFG 38), homeported in Yokosuka, Japan, from March 1996 until October 1997. In November 1997, he returned to

Washington, DC, and reported as Chief of Staff to the Program Executive Office, Theater Surface Combatants (PEO TSC), and then as Director, Battle Force Operations and Engineering in PEO TSC. He served as the Major Program Manager for the Rolling Airframe Missile (RAM)/Phalanx Close-in Weapon System (CIWS) Program Office (PMS 472) from December 1998 through March 2002. He was then assigned to PEO Integrated Warfare Systems as the Major Program Manager, Command and Control (Networks/Excomm) in March 2002. In this position, he directed and managed the ACAT ID Cooperative Engagement Capability (CEC) Program, all Navigation, Displays and Processors programs, and track management and sensor netting initiatives. Selected for Flag in April 2005, Frick served as Program Executive Officer for Integrated Warfare Systems from July 2005 to August 2007. He is currently assigned as Vice Commander Naval Sea Systems Command, Washington, DC.

Frick's personal awards include the Legion of Merit (with two Gold Stars), the Bronze Star (with Combat "V"), the Meritorious Service Medal (with two Gold Stars), the Navy and Marine Corps Commendation Medal (with three Gold Stars), the Navy and Marine Corps Achievement Medal, the Combat Action Ribbon, the German Iron Cross in Silver and various service and campaign Medals and Ribbons.

Modeling the Integration of Open Systems and Evolutionary Acquisition in DoD Programs

Presenter: David N. Ford received BS and MS degrees from Tulane University and his PhD degree from the Massachusetts Institute of Technology, Cambridge. He is a Professor in the Construction Engineering and Management Program, Zachry Department of Civil Engineering, Texas A&M University and a Research Associate Professor of Acquisition with the Graduate School of Business and Public Policy at the US Naval Postgraduate School in Monterey, CA. For over 14 years, he designed and managed the development of constructed facilities in industry and government. His current research interests include the dynamics of development supply chains, strategic managerial flexibility, and resource allocation policies.

David N. Ford, PhD, PE
Zachry Department of Civil Engineering
Texas A&M University
College Station, TX 77843-3136
Voice: (979) 845-3759
Fax: (979) 845-6554
Email: davidford@tamu.edu

Author: COL John T. Dillard, US Army (Ret.), managed major weapons and communications programs for most of his 26-year career in the military, including development efforts for the Army Tactical Missile System (ATACMS) and Javelin missile systems. His last assignment was head of all Defense Department contract administration in the New York metropolitan area. Dillard now serves as a senior lecturer with the Graduate School of Business and Public Policy at the US Naval Postgraduate School in Monterey, CA. He is also a 1988 graduate of the Defense Systems Management College, Program Managers course. His current research interests include defense acquisition policy, organizational design and control, and product development. He is a member of PMI and other professional organizations.

John T. Dillard
Graduate School of Business and Public Policy
Naval Postgraduate School
1 University Circle
Monterey, CA 93943
Voice: (831) 656-2650
Mobile: (831) 277-3457
Email: jdillard@nps.edu

Abstract

Open systems and evolutionary acquisition are two recent innovations designed to improve program performance with flexibility. The full potential of these approaches has not been captured, partially because of integration challenges during implementation. The current work investigates the impacts of open systems and evolutionary acquisition on DoD development programs. Development changes required to simultaneously use open systems and evolutionary acquisition are used to identify and describe impacts of implementation on program process and management. A dynamic simulation model of a program using both evolutionary acquisition and open systems is described and used to map these impacts. Simulation results generally support the suggested impacts identified in the literature and provide a possible explanation for changes in program performance. Implications for practice relate to changes in the types and timing of risk and a potential trading of design obsolescence risk for standards obsolescence risk.

Introduction

In order for the military to prepare to meet current and future capability demands, major DoD systems must improve with evolving technologies and be interoperable. The continued and, in some cases, accelerating evolution of technologies creates new challenges that are difficult to forecast and require rapid acquisition response. Integrated human-computer decision-making tools, advanced materials, NCS tools, and nano-level structures are examples of evolving technologies that present challenges and potential solutions that must be integrated by defense acquisition programs. The use of legacy and other weapon platforms, joint service solutions, the information and communication needs of Network Centric Systems (NCS), and coordination with allies in joint operations each require weapon systems that can operate across system, platform, and systems-of-systems boundaries. Providing those systems is an important acquisition challenge. Past DoD acquisition approaches have not fully provided the interoperability needed to meet these demands.

Open systems (OSJTF, 2004, September) and evolutionary acquisition (DoD, 2004, November, section 4.4.1) are two relatively recent DoD acquisition initiatives that seek to address system interoperability and technology evolution challenges and that help the DoD meet current and future capability needs. An open systems (OS) approach and evolutionary acquisition (EA) share several high-level objectives. Both approaches seek to improve performance over the system's lifetime and reduce acquisition cycle-time. Both approaches also attempt to improve system performance via flexibility for the integration of new technologies and information into systems as they evolve. The open systems approach facilitates upgrades through modularity. EA does this by multiple product releases and deliberate deferral of some functionality—allowing technologies and requirements to evolve and mature. Both OS and EA seek to reduce acquisition cycle-time to provide currently available functionality. OS provide a means of incorporating current and future functionality, and evolutionary acquisition limits the scope of development blocks to only the technologies and capabilities that are attainable in the near future.

Open systems and evolutionary acquisition share at least two important implementation approaches. First, both OS and EA incorporate flexibility into acquisition to manage uncertainty in technology. Open systems build flexibility into development products with modular design and standardized key interfaces. Evolutionary acquisition builds flexibility into development processes through the design of incremental capability blocks. These flexibilities create options that potentially increase system performance, reduce cost, or both, by allowing technological uncertainties to partially resolve before important development decisions are made. Second, both OS and EA place emphasis upon interfaces to address interoperability. Within an evolutionary approach, interface management is critical to successfully integrating designs across development blocks. This need increases for systems with interfaces across platforms or systems-of-systems. In contrast to these challenges, an OS approach focuses on explicitly identifying and managing key interfaces that can benefit from modular design and open systems as a means of improving interoperability.

The evolutionary acquisition challenge and the open systems method suggest that the two acquisition approaches must be integrated and may be synergistic. But the complexity of the processes and the requirements of the two approaches make their integration, synergy, and successful implementation anything but easy or certain. The requirements of the approaches have been largely identified, and some of the required changes in programs for the use of EA and OS together have been identified. But a focused study of the impacts of integrating open systems and evolutionary acquisition is needed both to identify the impacts on development processes and to point to potential program design and management actions in order to exploit their potential. It is not obvious how to investigate and better understand the integration and implementation issues presented by evolutionary acquisition and open systems as concurrent approaches. ***How does the use of evolutionary acquisition and open systems together impact a system's development processes and management? How do those impacts affect acquisition program performance?***

The current work partially addresses these issues as follows. The researchers review the evolutionary acquisition and open systems approaches through the lens of their influence on program processes and management. Required changes in programs identified in the existing literature are then used to describe challenges to integrating the approaches and to describe specific influences on program management. After describing the modeling approach and simulation model of an acquisition program, the researchers map the specific influences into changes in model variables. They then use the results of simulations of the evolutionary acquisition program without and with open systems as a basis for a discussion of both the needs for successful programs that use both approaches, as well as the use of simulation modeling as a tool for investigating these acquisition-implementation issues. The paper closes with recommendations for future work.

Evolutionary Acquisition

In the year 2000, the Defense Department promulgated the term "evolutionary acquisition" (EA) in its policy documents governing the strategy for acquisition of materiel and mandated such strategies be used as the preferred approach to procurement (USD(AT&L), 2000, October 23). Later elaborated as spiral and incremental strategies, these approaches contrast to others that are based on more serial, sequential or singular efforts to arrive at a product solution. The latter are often termed as: single-step-to-full-capability, grand design, big bang, technological leap, waterfall, rational-comprehensive, and the unified development method (Forsberg, Mooz & Cotterman, 2005, p. 354). The overarching goals and principles of the DoD's evolutionary acquisition are to ensure that the Defense Acquisition System provides useful military capability to the operational user as rapidly as possible, and such strategies shall be the preferred approach to satisfying operational needs. Evolutionary acquisition strategies define, develop, and produce/deploy an initial, militarily useful capability ("Block 1") based upon proven technology, time-phased requirements, projected threat assessments, and demonstrated manufacturing capabilities. They also plan for subsequent development and production/deployment of increments beyond the initial capability over time (Blocks II, III, and beyond) (USD(AT&L), 2000, October 23). Figure 1 shows the conceptual difference between a traditional single-step-to-capacity acquisition process and an evolutionary acquisition process with two development blocks, as described in the 1996 and 2003 versions of the *DoD 5000* series.

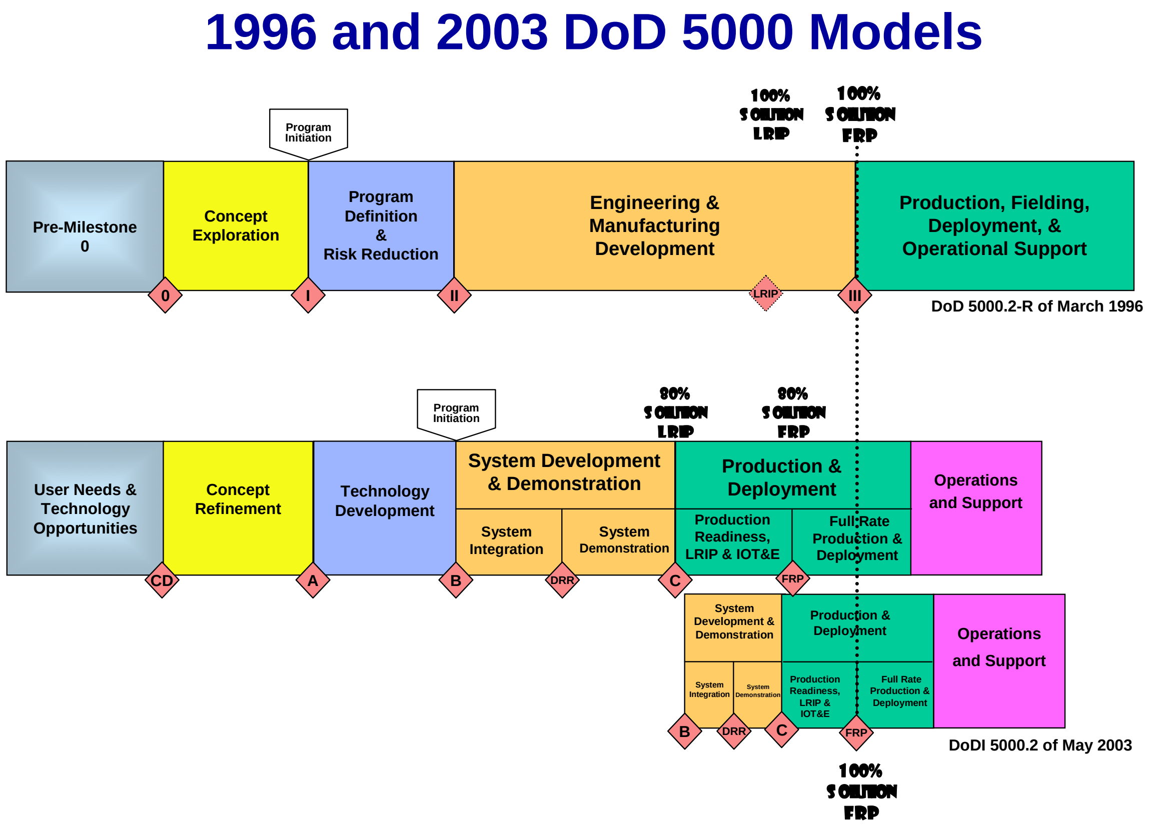

1996 and 2003 DoD 5000 Models
100% SOLUTION LRIP
100% SOLUTION FRP
Program Initiation
Pre-Milestone 0
Concept Exploration
Program Definition & Risk Reduction
Engineering & Manufacturing Development
Production, Fielding, Deployment, & Operational Support
0
I
II
LRIP
III
DoD 5000.2-R of March 1996
80% SOLUTION LRIP
80% SOLUTION FRP
Program Initiation
User Needs & Technology Opportunities
Concept Refinement
Technology Development
System Development & Demonstration
System Integration
System Demonstration
Production & Deployment
Production Readiness, LRIP & IOT&E
Full Rate Production & Deployment
Operations and Support
CD
A
B
DRR
C
FRP
System Development & Demonstration
System Integration
System Demonstration
Production & Deployment
Production Readiness, LRIP & IOT&E
Full Rate Production & Deployment
Operations and Support
B
DRR
C
FRP
DoDI 5000.2 of May 2003
100% SOLUTION FRP

Figure 1. Comparison of Traditional Single Step to Capacity and Evolutionary Acquisition Approaches
(Dillard, 2005)

The policy for evolutionary acquisition was aimed at improving all parameters of program success, but clearly and explicitly, its single most important objective was to reduce long product cycle-times to deliver operationally useful equipment. Figure 1 illustrates the hypothetical earlier start of production and the overlapping development blocks that are characteristic of evolutionary acquisition.

The authors, in their previous work (Dillard & Ford, 2007) investigated implementation challenges of evolutionary acquisition using the same approach we are using in the current work. We found, in part, that an evolutionary development approach significantly increases the number of development phases and activities that must be managed and coordinated at any given time over that required for single-block development. This, consequently, increases the organizational project management resource needs for successful acquisition over those necessary for single-block projects. Using open systems with an evolutionary approach may or may not accentuate these challenges.

Open Systems in DoD Acquisition

Open Systems were made a part of DoD acquisition in *DoD 5000.1* (USD(AT&L), 2003, May 12) which says, "a modular open systems approach shall be employed where feasible" (p. 7). A subsequent memorandum (USD(AT&L), 2004, July 7) clarified the central role of OS in acquisition by saying the approach is "an integral part of the toolset that will help DoD achieve its goal of providing the joint combat capabilities required in the 21st century, including supporting and evolving these capabilities over their total life-cycle" (p. 8). The Open Systems Joint Task Force (OSJTF) leads the DoD OS effort (OSTJF, 2004, September). Several terms defined in that guide are relevant to and used in the current work, including:

- **Open architecture:** An architecture that employs open standards for key interfaces within a system.
- **Open standards**: Standards that are widely used, readily available, consensus-based, published and maintained by recognized industry standards organizations (versus "closed," which are not).
- **Open system:** A system that employs modular design, uses widely supported and consensus-based standards for its key interfaces, and has been subjected to successful validation and verification tests to ensure the openness of its key interfaces.
- **Open systems environment (OSE):** A comprehensive set of interfaces, services, and supporting formats, plus aspects of interoperability of application, as specified by Information Technology (IT) standards and profiles. An OSE enables information systems to be developed, operated, and maintained independent of application-specific technical solutions or vendor products.

An open systems approach uses the concepts of key versus non-key interfaces and open versus closed interfaces, as defined above, to build flexibility into programs. Figure 2 illustrates potential locations of these interfaces in a conceptual system with modular subsystems/components. The centrality of these concepts to the open systems approach greatly increases the importance of the intended and unintended impacts of a shift away from the traditional focus on customized designs to integration through open interfaces.

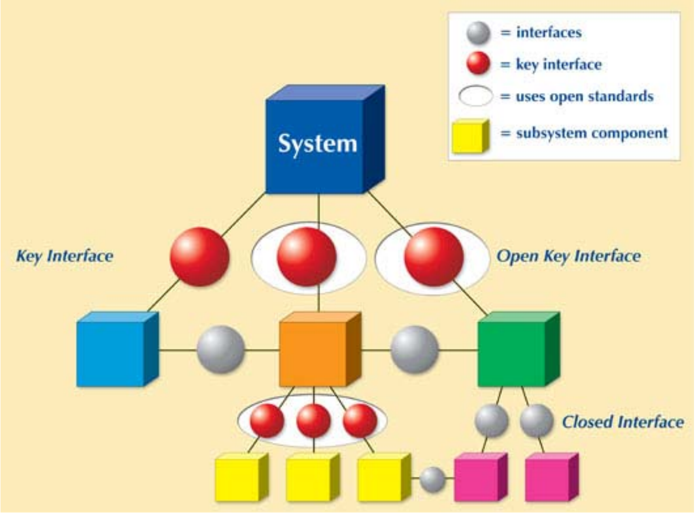


Figure 2. Types of Systems Interfaces
(OSJFT, 2004)

Challenges of Integrating Evolutionary Acquisition and Open Systems

Program managers using open systems and evolutionary acquisition in an integrated fashion may be able to achieve interoperability and insert evolving technologies better than project managers using either approach alone. But, despite their potential, the combination of OS and EA has not yet been fully developed or implemented in DoD acquisition. This is perceived to be largely because the issues related to their implementation have not been completely identified or resolved. This incomplete resolution of implementation for open systems and evolutionary acquisition individually makes understanding their interactions and the impacts of those interactions on acquisition programs difficult. Therefore, the challenges and solutions for implementing either approach are not yet fully understood. For example, the application of OS to hardware/software systems may present particular challenges.

The adoption and use of open systems in DoD acquisition requires several different activities that impact the acquisition process in different ways. Meyers and Oberndorf (2001) identify some of these activities. We describe the most important activities identified by Meyers and Oberndorf with our assessment of their impacts on the evolutionary acquisition process:

1. **Build a baseline of standards and commercial-off-the-shelf (COTS) products.** This change increases the scope of the Block 1 requirements phase and early design (pre-system acquisition) to describe the requirements in terms of standards.
2. **Build a high-level model of the system for use in applying the open systems approach.** This change increases the scope of early design in Block 1.
3. **Document the open architecture in a way that shows the evaluation of alternative architectures, identifies components, technologies, etc.** This change increases the scope of the early design activities and advanced development phases in all Blocks.
4. **Coordinate standards and establish liaisons with standards bodies and users.** This change increases scope of all phases in all blocks because it is an on-going process.
5. **Implement the use of the selected standards in the development process.** This change decreases scope of advanced development phase in all blocks due to component design activities being replaced with component selection.
6. **Integrate components into the product and test the integrated system.** This change increases problems/rework in advanced development and manufacturing phases of all blocks.

Hanratty, Lightsey, and Larson (1999) also investigated the use of open systems in acquisition. They describe the impacts of OS on acquisition as a shift away from design (which, in OS, is completed by the broader commercial market) to integration of elements into products (which, in OS, is increasingly completed with elements that were not developed specifically for the DoD). Hanratty, Lightsey, and Larson identified several areas of open systems design that pose risks, which we describe with our assessment of the primary impacts of OS on evolutionary acquisition processes.

1. **Slower integration and testing of standards-based elements into products.** This change delays the discovery of integration problems until later in projects.
2. **Reduced DoD control over standards.** This change increases the number and size of design problems due to faster evolution of the standard used in the product.
3. **Increased standards-selection risk due to evolution of standards and the possibility that standards will not endure.** This change both increases the number and size of design problems due to the possibility that the selected standard will not endure, and increases testing and integration (regardless of whether problems are discovered or not) due to more frequent changes in standards.
4. **Increased standard change risk—knowing when to shift from one standard to another.** This change increases testing and integration (regardless of whether problems are discovered or not) due to more frequent changes in standards. It also increases the number and size of integration problems that need to be discovered and resolved due to both the need to change to the new standard more often and the possibility of changing too early, too late, or to the wrong standard if more than one are available (e.g., competing for market dominance).
5. **Increased and continuous testing requirements due to the need to integrate evolving commercial and non-developmental items into systems.** This change increases testing and integration (regardless of whether problems are discovered or not) due to more frequent component redesigns.
6. **Development of support concepts early in the acquisition cycle—causing increased standards-selection risk due to large amounts of information needed about currently available standards.** This change increases standards research and planning early in acquisition, which would include increased interface design and management.
7. **Reduced control over detailed component design due to design by industry based on industry-controlled standards.** This change increases the number and size of integration problems due to component designs that do not exactly match product needs.

These specific influences pose significant individual challenges. However, they might also interact in ways that are difficult to predict or immediately recognize and address. In the Model Use section, we describe how we mapped these influences onto specific parts of an acquisition process to better understand how they impact program performance.

The Research Approach

Evolutionary acquisition and open systems approaches combine to create a complex set of development processes that evolve over time. An improved understanding of these processes and their management is available through formal modeling of the most important components and relationships that drive system performance and risk. Due to the number and complexity of the components and their relationships, the formal model structure and rigor of calculations can simulate and forecast performance and risk better than informal tacit predictions by humans. Therefore, we applied a computational experimentation approach to investigating evolutionary acquisition and open systems projects, integrating theory and practice in a computational tool that allows controlled experimentation through simulation. The current work reflects project, product development, and management theories.

The system dynamics methodology was applied to model a DoD acquisition project with evolutionary processes and open systems. System dynamics uses a computational experimentation approach to understanding and improving dynamically complex systems. The system dynamics perspective focuses on the roles of accumulations and flows, feedback, and nonlinear relationships in managerial control. The methodology’s ability to model many diverse system components (e.g., work, people, money), processes (e.g., design, technology development, quality assurance), and managerial decision-making and actions (e.g., forecasting, resource allocation) makes it useful for investigating acquisition projects. Forrester (1961) develops the methodology's philosophy, and Sterman (2000) specifies the modeling process with examples and describes numerous applications. When applied to development projects, system dynamics focuses on how performance evolves in response to interactions among development strategy (e.g., evolutionary development vs. traditional), managerial decision-making (e.g., scope developed in specific blocks), and development processes (e.g., concurrence). System dynamics is considered appropriate for modeling acquisition projects because of its ability to explicitly model critical aspects of development projects (Ford & Sterman, 1998; Cooper, 1993a,b,c; Cooper & Mullen, 1993; Cooper, 1994). System dynamics has been successfully applied to a variety of project management issues, including prediction/discovery of failures in project fast-track implementation (Ford & Sterman, 2003b), poor schedule performance (Abdel-Hamid, 1988), and the impacts of changes (Rodriguez & Williams, 1997; Cooper, 1980) and concealing rework requirements (Ford & Sterman, 2003a) on project performance. See Lyneis and Ford (2007) for a review of the application of system dynamics to projects.

The simulation model used here is based on previously developed system dynamics models of product development in several industries that have been developed and tested over several decades, as described and referenced below. Therefore, the model is founded on well-established and tested components. Previous models have developed structures for many components and aspects of acquisition. However, previous models have not been used to investigate the integration of EA and OS in acquisition projects. The current model was originally developed to investigate EA and is described in detail by Dillard and Ford (2007).

A Conceptual Model of an Evolutionary Acquisition Program

The model reflects the structure of development work moving through the separate development blocks of an acquisition project. In the model, four types of work flow through each block of an acquisition project: the development of requirements, the development of technologies, the design of product components, and the manufacture of products. Within a development block, each type of work flows through a development phase that completes a critical aspect of the project: 1) develop requirements, 2) develop technologies, 3) design product components (advanced development), and 4) manufacture products. The exception is requirements, which also measures progress through the final phase, 5) conduct user product testing. Development phases and information flows in a single block, as depicted in the model is shown in Figure 3. Arrows between phases indicate primary information flows. The start of all phases (except the development of requirements) is constrained by the completion of previous (“upstream”) phases. The completion of some requirements allows the start of technology development, reflecting the concurrent nature of this portion of acquisition. Both requirements development and technology development must be completed for Advanced Development to begin. The completion of Advanced Development allows manufacturing to begin. When some products have been manufactured, they are shipped to users for readiness testing. Figure 3 also identifies the five major reviews within a single acquisition block (A, B, Design Readiness Review, C, and Full-rate Production) at their approximate times during a project. These reviews are necessary, but are "off-core" activities that add work beyond that needed to complete the basic products of each phase (requirements, technologies, designs, products, and readiness for use confirmation).

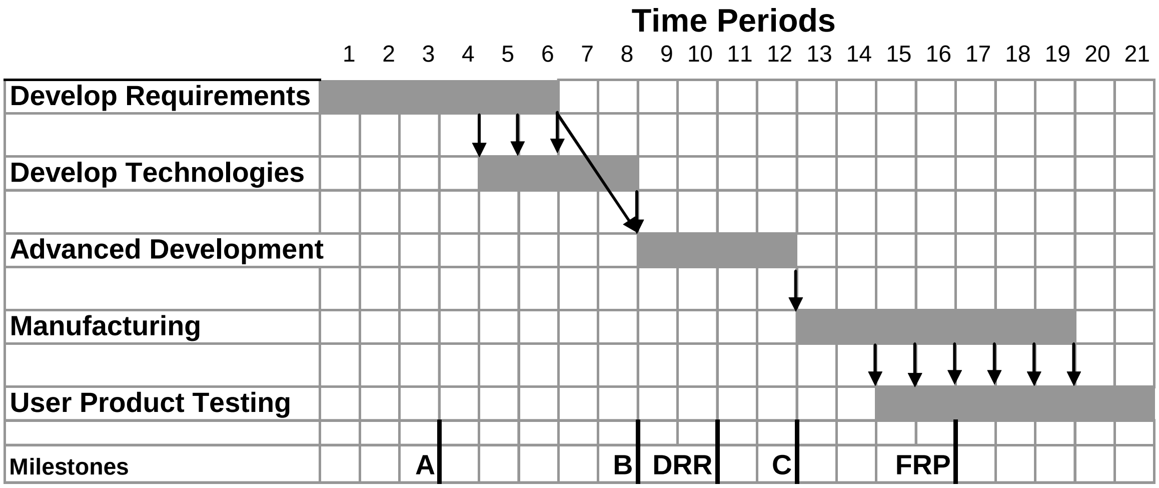


Figure 3. Information Flows in a Single Block Acquisition Project

Figure 4 depicts an acquisition project with multiple iterations or blocks. The first block is the same as Figure 3 above. Subsequent blocks have the same basic information flow, but can also be delayed by the completion of phases in previous blocks or constrained by the lack of progress in their own block. Importantly, in addition to the flow of information downstream through phases (black arrows in Figure 4), multiple iteration acquisition also provides opportunities for information to flow upstream, such as from User Product Testing in an earlier iteration to Develop Requirements or Advanced Development in a subsequent iteration (red vertical arrows in Figure 4).

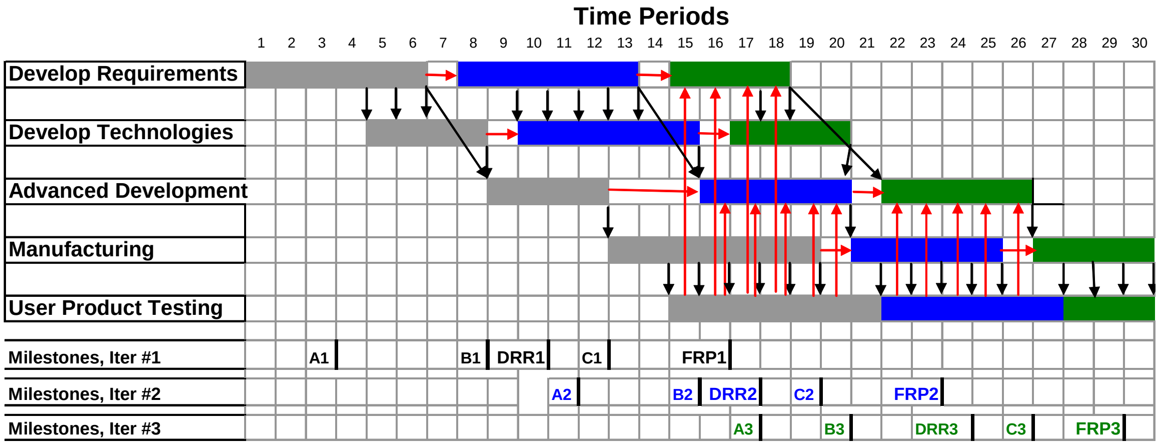


Figure 4. Information Flows in a Three-block Acquisition Project

A Formal Simulation Model of an Evolutionary Acquisition Program

The conceptual model described above was used to build a formal computer simulation model of an acquisition program that can reflect evolutionary acquisition and the use of open systems. See Dillard and Ford (2007) for details. The simulation model is a system of nonlinear differential equations. Each phase is represented by a generic structure, which is parameterized to reflect a specific phase of development.

Project performance is measured in three dimensions: schedule, cost, and product-performance risk. Schedule performance is measured by the time required to test and approve a given number or fraction of requirements by users. Cost is measured in dollars based on the size of direct and indirect work forces and the duration of phases and blocks. Product-performance risk is measured by the average percent of the requirements provided (approved by users) at any given time. This average reflects the combination of multiple requirements. All the requirements can be considered met completely when the average percent of the requirements provided is 100% for a development block.

The formal model was calibrated to the Javelin project described by Dillard and Ford (2007) based on data collected from a manager on the project (the second author) and performance data (e.g., schedule and costs) on the project. The model was tested with the three types of tests of system dynamics models suggested by Forrester and Senge (1980): structural similarity to the actual system, reasonable behavior over a wide range of input values, and behavior similarity to actual systems. The model was found to be useful for investigating the impacts of OS and EA on acquisition projects.

Model Use

To investigate the impacts of opens systems on evolutionary acquisition, we simulated a project similar to the Javelin project twice: first as if the project did not use open systems and then as if the project used an open systems approach. We then compared the behavior and project performance. The program base-case model and simulation described in Dillard and Ford (2007) reflects an evolutionary acquisition program that does not include open systems impacts. To add the impacts of open systems to the model, we first mapped the identified impacts based on Meyers and Oberndorf (2001) onto model variables as follows (Table 1):

Table 1. Impacts of Open Systems on Evolutionary Acquisition due to Changes Suggested by Meyers and Oberndorf (2001)

Change Required by Open Systems	Impact on Evolutionary Acquisition Processes
1) Build standards & COTS for program use	Increases Requirements scope in Block1
2) Build high-level model with open system	Increases Technology Development scope in Block 1
3) Document use of OS	Increases Technology Development scope in all blocks
4) Coordinate standards	Increases scope of all phases in all blocks

5) Implement OS	Decreases Advanced Development scope in all blocks
	Fewer Advanced Development design problems in all blocks
6) Integrate components	More Advanced Development integration problems in all blocks
	More Manufacturing integration problems in all blocks

We also mapped the impacts of required changes to acquisition projects identified by Hanratty, Lightsey, and Larson (1999) onto model variables as follows (Table 2):

Table 2. Impacts of Open Systems on Evolutionary Acquisition due to Changes Suggested by Hanratty, Lightsey, and Larson (1999)

Change Required by Open Systems	Impact on Evolutionary Acquisition Processes
7) Slower integration and testing	a1) Reduces problem discovery in Technology Development and Advanced Development phases in all blocks
	a2) Increases problem discovery in Manufacturing phases in all blocks
	b1) Decreases problem discovery in earlier blocks (all phases except Requirements)
	b2) Increases problem discovery in later blocks (all phases except Requirements)
8) Track and change with evolving standards	More problems in Advanced Development and Manufacturing phases in later blocks
	Increases scope in Technology Development and Advanced Development phases in all blocks
9) Increase testing to discover increased integration problems	Increases scope in Technology Development, Advanced Development, and Manufacturing phases in all blocks
10) Build support system (OSE)	Increased scope in Requirements phase in Block 1

Several of the changes above impact the same portions of an evolutionary process, sometimes in the same directions and sometimes in opposite directions. Therefore, we regrouped the impacts (Table 3) according to model variable that described a specific program block and development phase (e.g., scope of work in Block 1, Requirements Phase). The three variables found to best describe the impacts of open systems on evolutionary acquisition programs are the scope of work, rework fraction, and quality assurance (QA) effectiveness. In the table below and within the model, the scope represents the work that must be completed in a development phase. The Rework Fraction reflects the number of problems that are created in a development phase. The QA effectiveness reflects the difficulty of discovering problems to be resolved. The unit of measure of change was chosen as the percent change from the base case that the use of open systems would cause. This normalizes impacts for different phases (e.g., a change of 10 to a phase with a scope of 50 is very large compared to the same change to a phase with a scope of 5,000) and facilitates assessment of the changes. No known data is available to complete Table 3 based on an actual acquisition program. However, order of magnitude estimates that are in a reasonable rank order of size are adequate because of the preliminary nature of the study. The net changes of all the specific influences are summarized in Table 3. See Appendix A for a more detailed description of the estimates.

Table 3. Estimated Changes in Evolutionary Acquisition Processes to Reflect Open Systems

Program Block and Phase	Scope of Work	Rework Fraction	QA Effectiveness
DEVELOPMENT BLOCK 1			
Requirements	+7	0	0
Develop Techn.	-15	0	-10
Advanced Dev.	-17	-5	-10
Manufacturing	+2	+5	+5
Testing by User	+1	0	-5
Net Change from Base Case	**-0.22**	**0%**	**-20%**
DEVELOPMENT BLOCK 2			
Requirements	+1	0	0
Develop Techn.	-16	0	-5
Advanced Dev.	-17	0	-5
Manufacturing	+2	+10	+10
Testing by User	+1	0	0
Net Change from Base Case	**+29%**	**+10%**	**0%**
DEVELOPMENT BLOCK 3			
Requirements	+1	0	0
Develop Techn.	-16	0	0
Advanced Dev.	-17	+5	0
Manufacturing	+2	+15	+15
Testing by User	1	0	+5
Net Change from Base Case	**+29**	**+20**	**+20**

Simulation Results

Figure 5 shows a plot of the simulated percent of project requirements provided to users by the acquisition program without open systems (Line 1) and with open systems (Line 2). The simulated program has three development blocks, and the simulation clearly shows the evolutionary acquisition nature of the program—with three increases in requirements provided as each development block is completed. The simulation also shows the program with open systems provides as many or more requirements at any point in time than the program without open systems. This supports the open systems approach claim that it can facilitate providing more requirements faster.

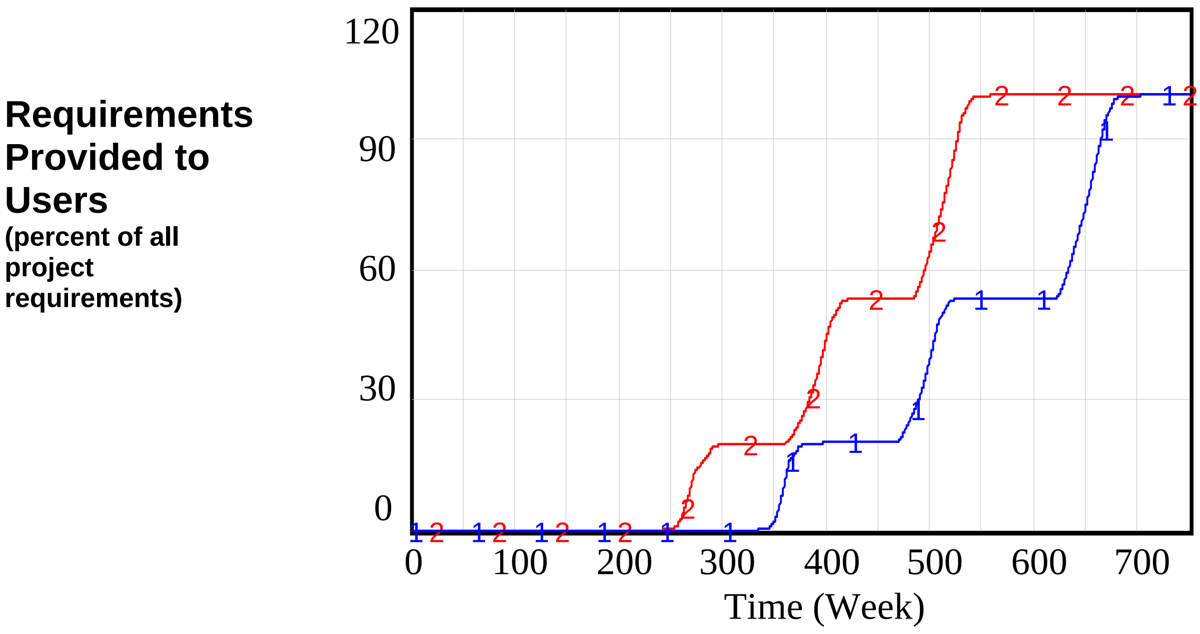


Figure 5. Requirement Fulfillment with Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

In addition to supporting the potential gains available through evolutionary acquisition and open systems, the simulation describes the interaction of evolutionary acquisition and open systems in more detail, providing the opportunity for improved understanding. The simulation shows that the improvement in time-to-requirement increases with each block, indicating that open systems can improve this dimension of program performance during multiple development blocks. An ***open systems approach may leverage its benefits when used with evolutionary acquisition through repeated capture of benefits generated in early development blocks in subsequent development blocks.*** If an OS approach is implemented with EA, programs may be able to reap the benefits first achieved in earlier blocks in subsequent downstream blocks, effectively benefitting more than once for the open systems work done early.

However, time to delivery of requirements is only one measure of program performance. Cost is another important performance measure. The simulated program without open systems costs $5.39 million through complete release to users, and the program with open systems costs $3.84 million through complete release to users.[1] Reduced costs are an established potential benefit of using open systems, largely through reduced design scope. This is the case in the model, in which a significant reduction in design scope is assumed to be a fundamental impact of using open systems. However, the simulation points out an additional potential cost benefit of using open systems. Shorter programs tend to cost less (all other things held equal). Therefore, ***open systems can***

[1] Actual cost savings may be significantly different due to smaller reductions in design scope.

improve cost performance by interacting with evolutionary acquisition to enhance the schedule performance available through evolutionary acquisition alone.

A third important performance measure is the quality of the developed product. Less-than-desired quality can be caused by many things, including not or partially fulfilling requirements, design errors that reduce product performance or increase operations or maintenance costs, and integration errors that make future upgrades difficult, slow, or expensive. Design and integration errors are particularly important in the current work because of their central role in open systems. Acquisition program changes required by open systems clearly alter the nature, number, and timing of both design and integration errors. Generally, early design errors are expected to be reduced, but later integration errors may increase due to evolving standards. Errors that are discovered and addressed during an acquisition program are not as problematic as those that remain after the product has been put into service. Undiscovered and released errors are problematic because they can severely increase operations, maintenance, and upgrade costs.

The model was used to simulate the number of undiscovered errors in released work without and with open systems. Figure 6 shows the evolution of the number of undiscovered and released errors as a percent of the program scope. In general, the number of released errors increases as work is completed until the next development phase begins receiving development work, finding errors, and returning them to upstream phases for resolution.

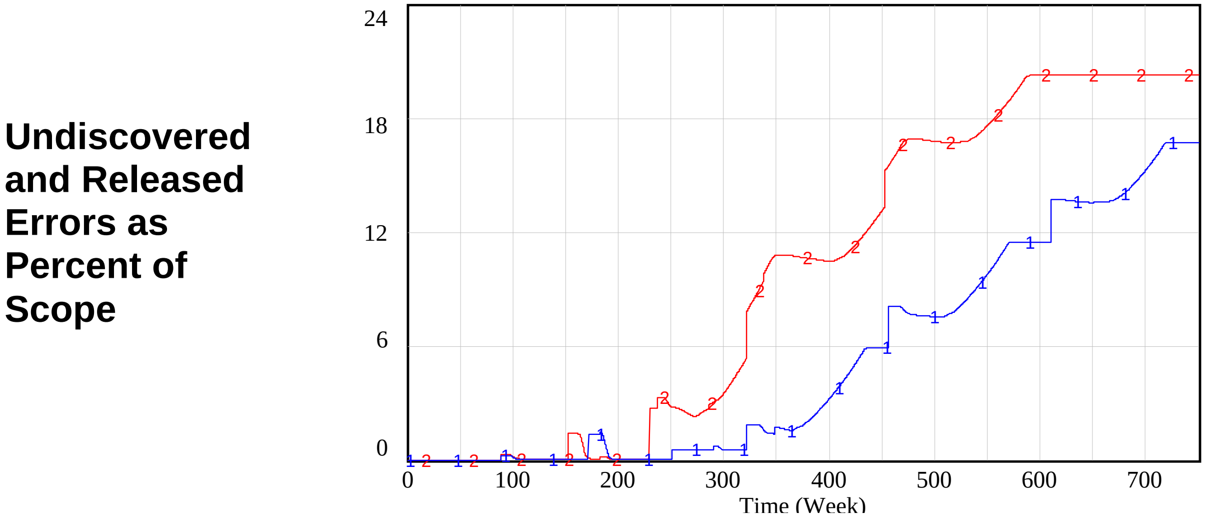


Figure 6. Undiscovered Problems in Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

Figure 6 shows that the simulated project with open systems generates and fails to find and resolve more errors before release. To further investigate this, the errors were disaggregated into design errors and integration errors based on the assumption that errors in the early development phases of each block (requirements and technology development and advanced development) are primarily design errors, and errors in manufacturing and testing are primarily integration errors. Figure 7 shows the undiscovered and released design errors as a percent of scope with and without open systems, and Figure 8 shows the undiscovered and released integration errors as a percent of scope with and without open systems. Note that the vertical scale in Figure 8 (0-20%) is four times larger than the vertical scale in Figure 7 (0-5%) for clarity.

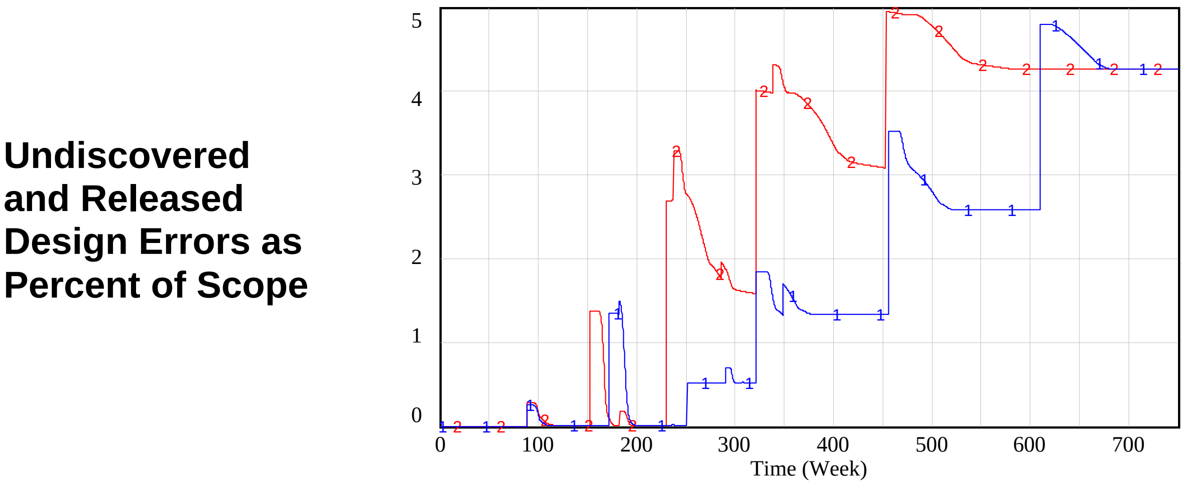


Figure 7. Undiscovered and Released Design Errors in Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

The differences in the timing of when design errors are generated, discovered and resolved, or missed and released is primarily due to the faster development with open systems. More importantly, the total percent of design errors at the completion of the program is nearly the same for the two programs. This suggests that the important impacts of open systems on evolutionary acquisition may be found in design errors.

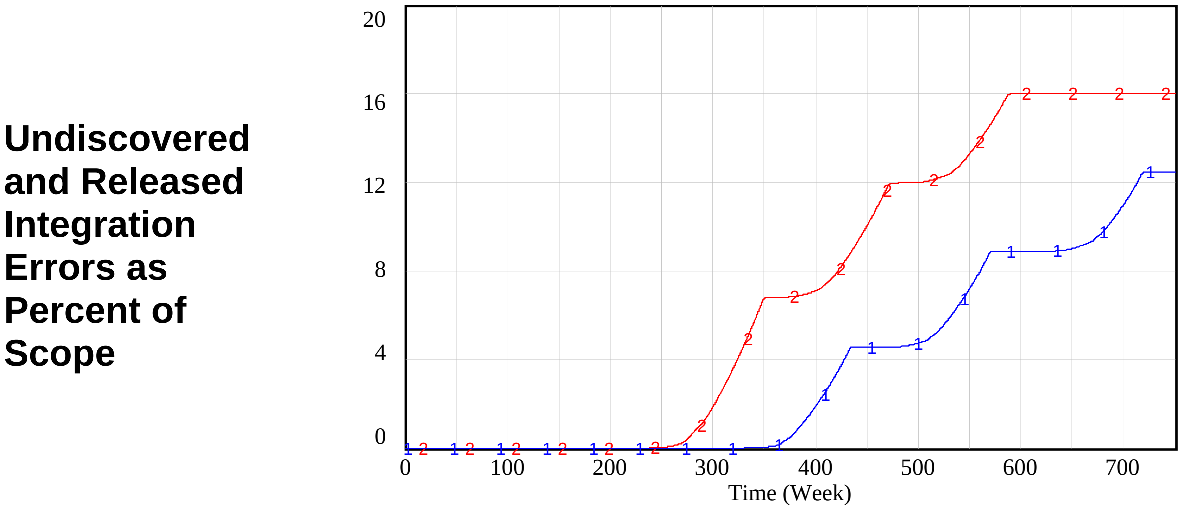


Figure 8. Undiscovered Integration Errors in Evolutionary Acquisition without (Line 1) and with (Line 2) Open Systems

There are at least two important differences between the number of undiscovered and released design errors (Figure 7) and the number of undiscovered and released integration errors (Figure 8). First, the programs generated and failed to resolve three to four times as many integration errors than design errors. This suggests that PMs using open systems must address integration issues if they wish to succeed. This finding also supports the importance of the shift from design to integration identified by other investigators. Second, the program with open systems generated at least 25% more integration errors than the program without open systems (3+% more than 13%). This difference in integration errors explains essentially the entire difference in total undiscovered and released errors (Figure 6).

In summary, the simulation results show that open systems can interact with evolutionary acquisition to improve the timing of products (Figure 5), reduce development costs, and increase the number of undiscovered and released integration errors (Figures 6-8). This suggests that ***open systems and evolutionary acquisition can interact to improve schedule and cost performance, but that these benefits may come at the cost of increased risk of high operations, maintenance, and upgrade costs when the integration errors are eventually discovered and must be resolved.***

Implications for Evolutionary Acquisition Practice with Open Systems

The identification of impacts of open systems on evolutionary acquisition programs and the simulation results carry potentially valuable implications for acquisition program managers.

Shifting the Types and Amounts of Risk

Adding open systems to evolutionary acquisition shifts the program management focus from design to standards and integration. This impacts when the program accepts and must manage different types and amounts of risk. Open systems reduce design risks by designing components, subsystems, and systems to be consistent with established standards. Design risk is also reduced, as an OS approach uses pre-designed and pre-tested components that have been designed and tested to established standards. Open systems may increase other risks, however. Standards selection and change risks are increased because programs using open systems are more dependent on standards than programs using customized designs; OS also have little influence over the evolution of those standards. Integration risks may increase significantly as standards change over the product lifecycle, and new standards may not be compatible with the current design of products. Different types of skills are needed to manage different types of risk. For example, detailed component design risk management requires technical expertise for design review and component testing, but integration risk management requires a broader, systems understanding of the product and how subsystems work together to fulfill requirements. Acquisition programs using open systems need a different set of risk-management skills than programs not using open systems. Less-detailed technical expertise will likely be needed, and more integration and systems expertise will be needed. ***If open systems are integrated into evolutionary acquisition (which repeats the development process over multiple blocks), then acquisition programs will require significant and extended integration and systems expertise.*** This will also change the skill sets needed by the DoD acquisition workforce.

A Temporal Shift in Program Risks

Design risks occur relatively early in programs and product lifecycles, whereas integration risks occur relatively late. Therefore, the use of open systems will shift program risk to emerge later in projects. The simulations support this result with the increase in the number of undiscovered and released integration errors with open systems. If costs follow risk, this may result in lower development costs due to lower design risk, but higher operating, maintenance, and upgrade costs due to higher integration risk. Figure 9 describes the relative costs in a product lifecycle. Integration of OS into EA may reduce Research and Development costs when programs can capture design benefits, but may increase Operating and Support costs when integration and evolving standards risks may increase costs. The sizes of these cost changes are uncertain, but the potential for early reductions in cost and later increases in cost are real.

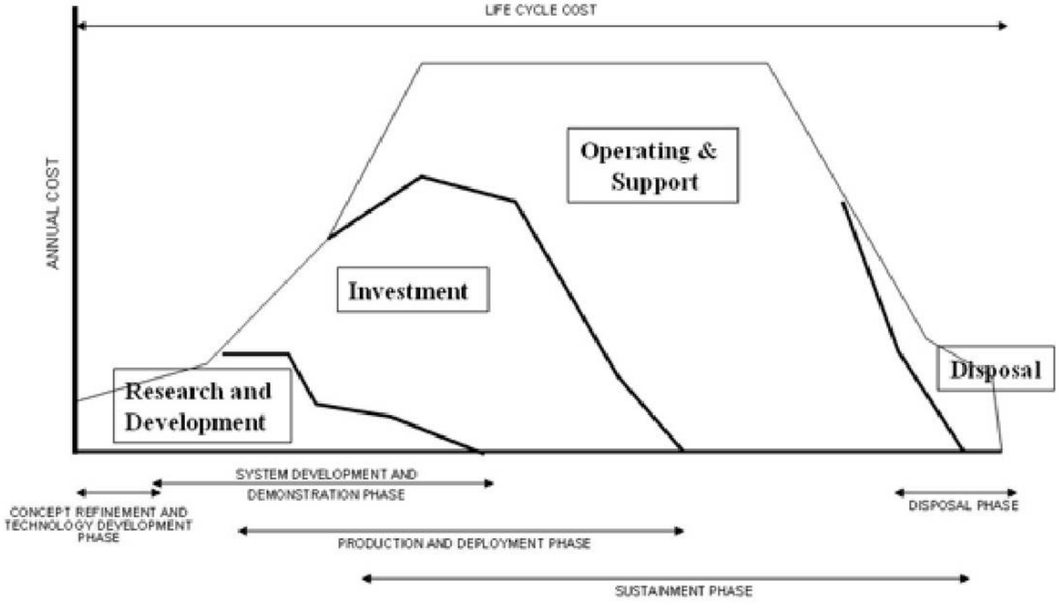

LIFE CYCLE COST
ANNUAL COST
Operating & Support
Investment
Research and Development
Disposal
SYSTEM DEVELOPMENT AND DEMONSTRATION PHASE
CONCEPT REFINEMENT AND TECHNOLOGY DEVELOPMENT PHASE
DISPOSAL PHASE
PRODUCTION AND DEPLOYMENT PHASE
SUSTAINMENT PHASE

Figure 9. Relative Costs during a Product Lifecycle
(Defense Acquisition Guidebook, 2004, November, p. 43)

By stretching acquisition across multiple blocks, evolutionary acquisition may accentuate the impacts of a temporal shift in program risk. Therefore, if using open systems causes this temporal shift in risks, then programs integrating open systems and evolutionary acquisition may experience an increase in the relative size of product costs during use.

Trading Design Obsolescence for Integration Obsolescence

Traditional acquisition processes commit programs to customized designs and, therefore, bear significant design obsolescence risk when threats and technologies evolve away from the design. An open systems approach can reduce that risk by allowing the use of more plug-and-play components that can be replaced with improved components that meet the chosen standard. However, by using open systems, a program must also commit to one or more standards early in development and, therefore, bear significant standards obsolescence risk if and as standards evolve away from the needs of the program and as integration problems increase. ***Evolutionary acquisition's need for integration across multiple development blocks can increase the impact of open systems on obsolescence risk. Adding open systems to evolutionary acquisition may cause programs to trade away design risk for increased integration risk.***

Conclusions

The current work has extended and expanded the descriptions of the impacts of using open systems and evolutionary acquisition together on development processes and management. We then mapped those impacts into a computer simulation model and used that model to investigate how open systems and evolutionary acquisition interact. Results include that the changes required to implement open systems in evolutionary acquisition significantly impact development processes and management, particularly scopes of design, standards, and integration work, the generation of different types of problems, and the timing of the discovery of problems. The shift from a focus on design to a focus on integration was found to be particularly important. Simulation reinforced the potential for open systems to accelerate acquisition and revealed a potentially important distinction between design and integration errors in explaining the impacts of required changes. Implications for practice included shifts in the type and timing of risks due to open systems use and the possibility of trading design obsolescence for integration obsolescence (e.g., compatibility).

This research has contributed to the understanding of open systems and evolutionary acquisition is several ways. The work improved the description and specification of impacts of acquisition policy on acquisition practice. The work also used dynamic computer simulation to model and investigate open systems and to model evolutionary acquisition and open systems together, both for the first time to our knowledge. The results of the simulation reinforced several suggested impacts of open systems and provided additional causal rationale behind why suggested impacts may occur. These rationales were the basis of potential implications for the evolutionary acquisition practice with open systems. The reasoning provided based on the computer simulation can be used to extend and deepen decision-makers' understanding of open systems and evolutionary acquisition and design program processes and management.

Future researchers can improve and extend the work described here by gathering additional data about the use of open systems with evolutionary acquisition in practice and, in so doing, testing the existence and importance of suggested impacts. The similarity of the model and, thereby, confidence in results can be improved by using additional acquisition projects that use both evolutionary acquisition and open systems.[2] Finally, additional recommendations for practice can be developed based upon the model developed here and elsewhere. These investigations can further develop the understanding of how to effectively integrate open systems and evolutionary acquisition and, consequently, improve the systems and products provided to warfighters.

[2] The authors are currently working with a large navy acquisition project to do this.

List of References

Abdel-Hamid, T. (1988). Understanding the “90% syndrome” in software project management: A simulation-based case study. *The Journal of Systems and Software*, *8*, 319-330.

Cooper, K.G. (1980). Naval ship production: A claim settled and a framework built. *Interfaces*, *10*(6), 20-36.

Cooper, K.G. (1993a, February). The rework cycle: Why projects are mismanaged. *PM network*, 5-7.

Cooper, K.G. (1993b, February). The rework cycle: How it really works…and reworks. *PM network*, 25-28.

Cooper, K.G. (1993c). The rework cycle: Benchmarks for the project managers. *Project Management Journal*, *14*(1), 17-21.

Cooper, K.G. (1994). The $2,000 hour: How managers influence project performance through the rework cycle. *Project Management Journal*, *15*(1), 11-24.

Cooper, K.G., & Mullen, T. (1993). Swords and plowshares: The rework cycles of defense and commercial software development projects. *American Programmer*, *6*(5), 41-51.

Dillard, J.T. (2005, August-November). Toward centralized control of defense acquisition programs: A comparative review of the framework from 1987 – 2003*. Acquisition Review Journal, 12*(3), 331-344.

Dillard, J.T., & Ford, D.N. (2007, April 30). *From amorphous to defined, balancing the risks of evolutionary acquisition* (NPS-AM-07-002). Monterey, CA: Naval Postgraduate School.

DoD. (2004, November). *Defense acquisition guidebook*. Retrieved April 4, 2008, from https://akss.dau.mil/DAG/Guidebook/IG_c4.4.1.asp)

Ford, D., & Sterman, J. (1998, Spring). Modeling dynamic development processes. *System Dynamics Review*, *14*(1), 31-68.

Ford, D., & Sterman, J. (2003a, September). The liar's club: Impacts of concealment in concurrent development projects. *Concurrent Engineering Research and Applications*, *111*(3), 211-219.

Ford, D., & Sterman, J. (2003b, September). Overcoming the 90% syndrome: Iteration management in concurrent development projects. *Concurrent Engineering Research and Applications*, *111*(3), 177-186.

Forrester, J.W. (1961). *Industrial dynamics*. Cambridge, MA: MIT Press.

Forrester, J., & Senge, P. (1980). Tests for building confidence in system dynamics models. *TIMS Studies in the Management Sciences*, *14*, 209-228.

Forsberg, K., Mooz, H., & Cotterman, H. (2005). *Visualizing project management* (3rd ed.). Hoboken, NJ: John Wiley & Sons.

Hanratty, M., Lightsey, R.H., & Larson, A.G. (1999, Winter). Open systems and the systems engineering process. *Acquisition Review Quarterly, 6*(4), 47-59.

Lyneis, J.M., & Ford, D.N. (2007). System dynamics applied to project management: A survey, assessment, and directions for Future Research. *System Dynamics Review, 23*(4), 157-189.

Meyers, B.C., & Oberndorf, P. (2001). *Managing software acquisition: Open systems and COTS products.* Boston, MA: Addison-Wesley.

Open Systems Joint Task Force (OSJTF). (2004, September). *Open systems joint task force (OSJTF) program manager’s guide: A modular open systems approach (MOSA) to acquisition*. Retrieved April 9, 2008, from http://www.acq.osd.mil/osjtf/mosapart.html.

Rendon, R.G. (2006, January). *Using a modular open systems approach in defense acquisitions: Implications for the contracting process* (NPS-GSBPP-06-007). Monterey, CA: Naval Postgraduate School.

Rodriguez, A., & Williams, T.M. (1997). System dynamics in project management: Assessing the impacts of client behavior on project performance. *Journal of the Operational Research Society*, *49*, 2-15

Sterman, J.D. (2000). *Business dynamics, system thinking and modeling for a complex world.* New York: Irwin McGraw-Hill.

USD(AT&L). (2000, October 23). *Operation of the defense acquisition system* (DoD Directive 5000.2). Washington, DC: Author.

USD(AT&L). (2003, May 12). *The defense acquisition system* (DoD Directive 5000.1). Washington, DC: Author.

USD(AT&L). (2004, April 5). *Amplifying DoDD 5000.1. Guidance regarding modular open systems approach (MOSA) implementation.* Memorandum. Washington, DC: Author.

USD(AT&L). (2004, July 7). *Instructions for Modular Open Systems Approach (MOSA) implementation*. Memorandum. Washington, DC: Author.

Appendix A. Mapping Specific Influences of Open Systems onto Evolutionary Acquisition Programs Processes

The researchers estimated the impact of each specific, identified and described influence on the scope of work, rework fraction, and quality assurance (QA) effectiveness. They measured the scope of work by the number of equal-sized work packages that must be completed in a development phase. They measured the rework fraction by the percent of those work packages that require changes; this measurement reflects the number of problems that are created in a development phase. They measured the QA effectiveness with the fraction of the work packages discovered to need rework. Although no known data is available as a basis for the estimated changes, order of magnitude estimates that are in a reasonable rank order of size are adequate because of the preliminary nature of the study. To facilitate mapping of the specific influences above to model changes, the researchers listed the specific influences after the individual impacts on each model parameter.

Table 4. Detailed Estimated of Changes in Evolutionary Acquisition Processes to Reflect Open Systems

Program Block and Phase	Scope of Work	Rework Fraction	QA Effectiveness
DEVELOPMENT BLOCK 1			
Requirements	+1+1+5 (1,4,10) +1+1+1-20	0	0
Develop Techn.	+1+1(1,2,3,5,8,9)	0	-5 -5 (7a,7b)
Advanced Dev.	+1-20 +1+1 (4,5,8,9)	-10 +5(5,6)	-5 -5 (7a,7b)
Manufacturing	+1 +1(4,9)	+5 (6)	+10 -5 (7a,7b)
Testing by User	+1 (4)	0	-5 (7b)
Net Change in Base Case	**-22**	**0**	**-20**
DEVELOPMENT BLOCK 2			
Requirements	+1 (4)	0	0
Develop Techn.	+1+1 -20+1+1 (3,4,5,8,9)	0	-5 (7a)
Advanced Dev.	+1-20 +1+1 (4,5,8,9)	-10 +5 +5(5,6,8)	-5 (7a)
Manufacturing	+1 +1(4,9)	+5 +5 (6,8)	+10 (7a)
Testing by User	+1 (4)	0	0
Net Change in Base Case	**29**	**10**	**0**
DEVELOPMENT BLOCK 3			
Requirements	+1 (4)	0	0
Develop Techn.	+1+1-20+1+1 (3,4,5,8,9)	0	-5 +5 (7a,7b)
Advanced Dev.	+1-20+1 +1(4,5,8,9)	-10 +5+10 (5,6,8)	-5 +5 (7a,7b)
Manufacturing	+1+1 (4,9)	+5 +10 (6,8)	+10 +5 (7a,7b)
Testing by User	+1 (4)	0	+5 (7b)
Net Change in Base Case	**29**	**20**	**20**

THIS PAGE INTENTIONALLY LEFT BLANK

Panel 5 - Technology Maturity Considerations in Defense Acquisition

Wednesday, May 14, 2008

1:45 p.m. – 3:15 p.m.

Panel 5 - Technology Maturity Considerations in Defense Acquisition

Chair:

Dr. Michael F. McGrath, Vice President, Systems and Operations Analysis, Analytic Services, Inc.

Discussant:

Dr. James Snider, MG, USA (Ret.), Associate Vice President for Research, University of Alabama in Huntsville

Papers:

The Costs and Risks of Maturing Technologies, Traditionally vs. Evolutionary Approaches

Dr. William B. Rouse, Executive Director, Tennenbaum Institute and **Dr. Michael J. Pennock**, Research Fellow, Tennenbaum Institute, Enterprise Transformation, Georgia Institute of Technology

System Maturity Indices for Decision Support in the Defense Acquisition Process

Dr. Brian J. Sauser, Assistant Professor and **Dr. Jose E. Ramirez-Marquez**, Assistant Professor, Stevens Institute of Technology

Chair: Dr. Michael McGrath is the Deputy Assistant Secretary of the Navy for Research, Development, Test and Evaluation. His role is to aggressively drive new technologies from all sources across Navy and Marine Corps platforms and systems, and to develop programs to bridge the gap in transitioning new capabilities from science and technology (S&T) to acquisition.

Discussant: Dr. James Snider, MG, USA (Ret.), covers information policy at the New America Foundation, whose purpose is "to bring exceptionally promising new voices and new ideas to the fore of our nation's public discourse." Snider came to New America after a year in the United States Senate, where he served as an American Political Science Association Congressional Fellow in Communications and Public Policy. Snider is a graduate of Harvard College and holds graduate degrees in Political Science from Northwestern University and in Business Administration from the Harvard Business School. Snider is the co-author of *Future Shop*, one of the first books on the emerging area of e-commerce, and his papers on information policy have been reprinted in a number of other books. A former school board member, he has written articles on e-education that have appeared in *Education Week* and widely used McGraw-Hill textbooks, including *Computers in Education* and *Issues in Education*.

The Costs and Risks of Maturing Technologies, Traditionally vs. Evolutionary Approaches

Presenter: Michael Pennock is a research fellow at the Tennenbaum Institute for Enterprise Transformation. He has previously worked as a systems engineer for the Northrop Grumman Corporation where he developed and analyzed system architectures. Pennock earned his PhD in Industrial and Systems Engineering at Georgia Tech, and his research focuses on adapting economic analysis to address problems in systems engineering.

Michael J. Pennock, PhD
Research Fellow
Tennenbaum Institute
760 Spring Street, NW
Atlanta, GA 30332-0210
E-mail: mpennock@ti.gatech.edu

Author: Bill Rouse is the Executive Director of the Tennenbaum Institute at the Georgia Institute of Technology. He is also a professor in the College of Computing and School of Industrial and Systems Engineering. Rouse has written hundreds of articles and book chapters, and has authored many books, including most recently *People and Organizations: Explorations of Human-Centered Design* (Wiley, 2007), *Essential Challenges of Strategic Management* (Wiley, 2001) and the award-winning *Don't Jump to Solutions* (Jossey-Bass, 1998). He is editor of *Enterprise Transformation: Understanding and Enabling Fundamental Change* (Wiley, 2006), co-editor of *Organizational Simulation: From Modeling & Simulation to Games & Entertainment* (Wiley, 2005), co-editor of the best-selling *Handbook of Systems Engineering and Management* (Wiley, 1999), and editor of the eight-volume series *Human/Technology Interaction in Complex Systems* (Elsevier). Among many advisory roles, he has served as Chair of the Committee on Human Factors of the National Research Council, a member of the US Air Force Scientific Advisory Board, and a member of the DoD Senior Advisory Group on Modeling and Simulation. Rouse is a member of the National Academy of Engineering, as well as a fellow of four professional societies—Institute of Electrical and Electronics Engineers (IEEE), the International Council on Systems Engineering (INCOSE), the Institute for Operations Research and Management Science, and the Human Factors and Ergonomics Society.

William B. Rouse, PhD
Executive Director and Professor
Tennenbaum Institute
760 Spring Street, NW
Atlanta, GA 30332-0210
Phone: 404-894-2331
E-mail: bill.rouse@ti.gatech.edu

Abstract

Evolutionary acquisition holds the potential to improve both the cost of defense acquisition and the performance of acquired systems. Traditional acquisition programs tend to employ promising, yet immature, technologies and develop them within the program. Because immature technologies are inherently risky, unforeseen obstacles to development can lead to substantial cost overruns and schedule delays. This results in infrequent, but large, increments of deployed capability. In contrast, evolutionary acquisition employs more mature, less-risky technologies. This results in more frequent,

smaller increments of deployed capability. In theory, evolutionary acquisition could be more cost effective than traditional acquisition approaches because it avoids most of the risk inherent to technology development. However, there is a latent issue regarding evolutionary acquisition. If technology is not matured within a program, it must be matured somewhere else. For critical, DoD-specific technologies, this cost must logically fall on the DoD itself. The question, then, is whether it is more cost effective to mature technologies within the R&D system or within an acquisition program? A simulation of the defense acquisition system is developed to address this question.

1. Introduction

Over the past several years, the United States Department of Defense (DoD) has been attempting to transform itself from an organization designed to meet the Cold War threat of the Soviet Union to a more flexible, adaptable organization that is ready to meet the regional and asymmetric threats the US expects to face in the coming years. To facilitate this transformation, several modifications have been made to the defense acquisition system—the most important being the shift to evolutionary acquisition.

Evolutionary acquisition is an attempt to address one of the most common criticisms of the defense acquisition system. Traditional acquisition programs attempt large, revolutionary leaps in system capability through the use of immature and risky technology. Not only does immature technology often require more time and money to develop, but it also introduces uncertainty that may lead to significant delays and cost overruns. Consequently, warfighters must often make due with increasingly obsolete equipment during the long intervals between new system deployments, and there is little flexibility to adapt to emerging threats and exploit technology opportunities.

Evolutionary acquisition, on the other hand, attempts to set more modest capability goals for each acquisition. The idea is to use more mature, and hence, less-risky technology, in order to shorten acquisition cycle-times. Thus, each acquisition cycle under evolutionary acquisition should be shorter and cost less than more traditional programs. As a result, warfighters should receive more frequent upgrades to their equipment and, thus, should be at less risk of going to war with obsolete hardware.

Despite the apparent motivation to implement evolutionary acquisition and committing the approach to policy, it would seem that the DoD has had limited success in doing so (Lorell, Lowell & Younossi, 2006). In fact, the US Government Accountability Office (GAO) has suggested that DoD reforms have not gone far enough (GAO 2003; 2006, September; 2006, April; 2007a). They advocate adapting commercial best practices regarding technology and product development to the defense acquisition system. Among these are a centralized portfolio approach to managing new systems and technologies, a staged knowledge-based approach to both acquisition and technology development, strict enforcement of technology maturity requirements, and a more evolutionary approach to new system development.

Since these reforms are derived from the commercial world, the obvious question is whether they will translate well to a government context. The defense acquisition system differs from a commercial product development process in several respects. In particular, the government essentially serves as a technology developer, system developer, customer, and user. Furthermore, the DoD and a few allies are really the only customers for the systems and technologies developed within the defense acquisition system. Thus, there is a more limited capacity to purchase multiple evolutionary iterations of a system than there would be with a consumer product. Consequently, the pertinent question is, if evolutionary acquisition were fully implemented, would there be any tangible benefit for the Department of Defense? Since evolutionary acquisition is inseparable from technology policy, the objective of this paper is to consider the implications of an evolutionary technology policy on the cost and performance of the defense acquisition system.

As a first step to better understanding the system level trade-offs of technology policy on acquisition, the work presented in this paper attempts to model the basic "physics" of the acquisition system, in particular, the relationship between the R&D process and the acquisition lifecycle. The purpose is to gain insight into the most fundamental system-level influences on the efficacy of acquisition policies. To that end, an idealized view of the acquisition system is adopted to which complicating factors may be subsequently added to test their effects. The acquisition model was implemented as a discrete event simulation with the key decision variable being the maturity level at which a technology moves from R&D to an acquisition effort. Extensive sensitivity analyses were performed and several insights into the impact of technology policy on acquisition were generated. The most important output of this effort, however, is an informed set of future research directions that will facilitate more definitive answers to major policy questions regarding evolutionary acquisition. What follows is a summary of key findings. For a more detailed discussion of the analysis approach and results see Pennock (2008).

2. Background

As was mentioned previously, evolutionary acquisition is an attempt to reduce acquisition cycle-times by setting capability goals that are more modest than is typical of a traditional program. This allows programs to utilize more mature technology and, hence, reduce the amount of technology risk. In theory, this should reduce cost, schedule, and performance uncertainty. The hope is that it will lead to less-expensive acquisition programs that proceed more quickly. Consequently, warfighters would receive up-to-date equipment more frequently and at lower cost.

The motivating issue behind evolutionary acquisition is cycle-time. In theory, shorter cycles mean that each is less expensive and new technologies can be moved into the field faster to meet emerging warfighter needs. The driving issue, then, is how big of a leap in capability should one attempt during each acquisition cycle? Of course, the risk associated with the size of the leap is linked to the maturity of the required technology. Thus, evolutionary acquisition is really all about technology policy because with a large enough leap, evolutionary becomes revolutionary.

So where does the DoD's approach to evolutionary acquisition come in? A key issue is that the DoD does not manage technology or "product" portfolios in the same manner as a large commercial enterprise. In part, this is due the public nature of the defense enterprise. Even so, the GAO asserts that the DoD should adopt additional commercial best practices regarding the centralized management of its acquisition and technology portfolios and the management of technology transitions from R&D to acquisition programs (GAO, 2006, September; 2007a). Under the current system, there is often a funding gap in technology development. Early-stage technologies are funded through the R&D system (or S&T as it is known in the DoD), and late-stage technologies are often funded in support of a particular acquisition effort. It is technologies in the middle stages of maturation that are often left without obvious ownership and hence funding. Consequently, if certain technologies are required by an acquisition effort, their development through the middle stages must be funded in support of the development of a particular system. This requires early commitment to a technology when its final realization is still uncertain. In the past, this has often led to disappointment as technologies took longer to develop and did not perform as well as expected. In fact, a recent National Research Council report suggests that concept decisions made prior to Milestone A determine 70-75% of lifecycle costs (NRC, 2008). Early commitment to system concepts that depend on immature technologies sacrifices flexibility and may lead to costly rework.

Theoretically, if the DoD adopted the commercial new product model that the GAO suggests (GAO, 2006, September; 2007a), it would allow the DoD greater flexibility in how to select and mature technologies for development in anticipation of future acquisition program needs. In essence, it would lead to the creation of additional technology options. This would reduce the burden and risks of technology development on acquisition programs since they could choose from a portfolio of mature technologies.

So in the end, the two fundamental questions of evolutionary acquisition are how mature should technologies be when they are transitioned from R&D to acquisition efforts, and what is the best approach to mature them? All else being equal, this essentially determines the acquisition cycle-time as well as the size of the capability improvement for each cycle. Ultimately, the answer will hinge on factors such as the cost of technology maturation, the rate of learning from fielded systems, and the overhead cost associated with an acquisition cycle.

3. Model Setup

The motivation behind the structure of the model is to represent the set of commercial best practices recommended by the GAO for implementation in the context of the defense acquisition system. This includes both a staged, centrally managed technology development process as well a strictly enforced acquisition program lifecycle. Given the staged nature of both R&D and acquisition, discrete event simulation was the logical choice to capture the behavior of the system. As was mentioned previously, the representation of the defense acquisition system presented here is intentionally scaled-down and idealized. The benefit of an idealized model is two-fold. First, the scaled-down representation is more tractable and allows us to attempt multiple experimental excursions. Second, it allows us to consider the structural impacts of technology policy unobscured by the inconsistent implementation that occurs in the actual defense acquisition system. In particular, the modeling emphasis was on the linkage between the movement of technologies through the R&D process to the length and cost of the acquisition cycles. In order to represent the impact of technology policy on defense acquisition, there were three key features of the system that required consideration: the movement of technologies through the R&D system, the movement of programs through the acquisition process, and the rate of technological progression.

The simulation was implemented using the Arena 10.0 software package and consists of three major components: the technology development process model, the system acquisition process model, and the technical progress model. The technology development process model describes how technologies with potential defense application are matured through the defense R&D system. This process provides a portfolio of technologies for use by acquisition programs. The system acquisition process model describes the lifecycle of a defense acquisition program from concept development to deployment. Finally, the technical progress model describes how the capabilities provided by technologies improve over time.

3.1 Technology Development Process Model

The technology development process model simulates the movement of individual technologies through a maturation process. Ideally, a technology development process is centrally managed and staged. Technologies are selected for development based on their potential applicability to future products. In the commercial world, product and technology roadmaps drive development. These roadmaps, and the organization's commitment to them, provide a shared vision that the DoD often lacks. However, developing the technologies to satisfy the roadmap entails a certain amount of risk. In order to mitigate that risk, each technology must pass through a series of stage-gates. Each gate provides an opportunity to evaluate the status of a technology and determine whether it should continue to receive funding. Such a system facilitates prioritization of technology projects as well as risk mitigation. It is important to note that the Department of Defense has not consistently implemented such a system (GAO, 2006, September). Instead, there are a number of different organizations throughout the DoD that perform or fund R&D work, each with its own way of managing technology projects. These inconsistencies preclude the effective management of technology development and promote duplication and mismatch between the technology supplied by R&D organizations and the technology demanded by acquisition programs. Consequently, for this study, the technology development process was modeled in the spirit of the GAO's recommendation of a centrally-managed and staged technology development process.

The process starts when new but immature technologies arrive for evaluation. The arriving technologies are prioritized and then funded until the budget is expended. Rejected technologies are considered for funding in future rounds, and successfully matured technologies move on to the next stage. The sequence repeats until each technology is either successfully matured or discarded. The maturity of each technology is measured by the Technology Readiness Level (TRL) scale. The purpose of a properly functioning technology development process is to prioritize and fund these technologies by potential cost and benefit. The process used in this simulation is represented in Figure 1. For more information see Pennock (2008).

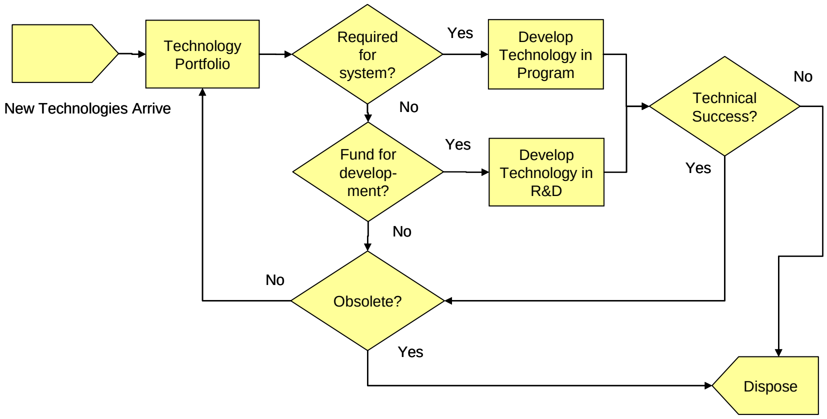


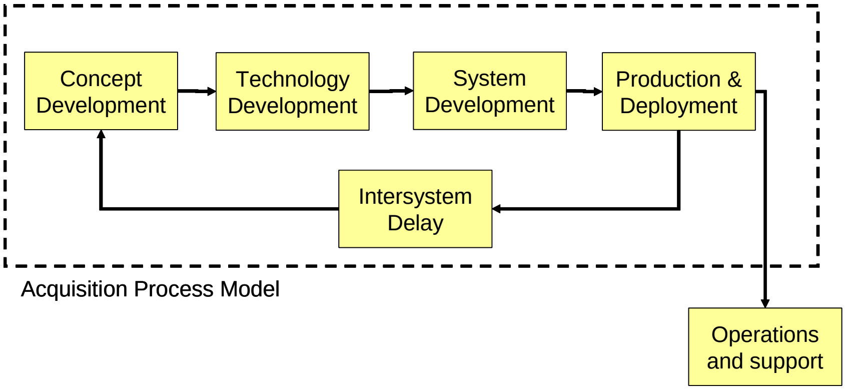
Figure 1. The Technology Development Process Model

3.2 System Acquisition Process Model

The system acquisition process model describes the lifecycle of a defense acquisition program. It is based on the five-stage Defense Acquisition Management Framework (DoD, 2006). As with the technology development process model described above, the acquisition process model in the simulation will assume that acquisition programs follow the rules, and, consequently, programs will move through each phase in order with no concurrency.

Within the simulation model, the basic unit in the system acquisition process is a program to acquire a system. It is assumed that the DoD has several types of systems. Each type is continuously cycling through the acquisition process. For example, if the Air Force deploys a new air superiority fighter, it is assumed that it will begin concept development of its replacement shortly thereafter. This assumption will be relaxed later.

Each type of system is dependent upon several technologies, each from a different application area. For example, an air superiority fighter might require a propulsion technology, a sensor technology, and an avionics technology. The acquisition process model used in the simulation is illustrated in Figure 2. For more information see Pennock (2008).



Note: Operations and Support was considered outside the scope of this simulation.

Figure 2. The System Acquisition Process Model

Ultimately, the purpose of acquiring a system is to provide military capabilities. It is assumed that each system deployed provides a capability. Capability in the model is an abstract representation of military utility. It is assumed that there is a synergistic effect between the technologies employed in the system: the system being greater than the sum of its parts. Thus, a multiplicative model is used to represent capability. The capability of a deployed system is the product of the performance levels for each of its required technologies. Thus, an air superiority fighter without a propulsion system is useless no matter how capable its sensor is. This measure of capability allows us to determine the cost effectiveness of a particular technology policy.

3.3 Technical Progress Model

The final key feature of the simulation is the model of technical progress. Where do new, more capable technologies come from? It is important to note that the technology development model in this simulation does not consider basic research. In fact, TRL 1 signifies the transition of ideas, concepts, and technologies from basic research to applied research. Thus, we can assume that there is a certain amount of research occurring exogenous to the simulation. This research may come from government or commercial sources. The key is that there is a constant inflow of new technologies and that their performance improves over time. The purpose of the technology development process is to adapt these technologies for use in a military system. There is one caveat, however, and that is that a purely exogenous technical progress model neglects the learning that inevitably occurs from fielding systems. For example, valuable information gathered from field use of a jet engine will likely inform the development of the next generation jet engine. Thus, there is a learning effect, and the more rapidly systems are fielded, the sooner subsequent learning will be available for future technologies. This is especially true for military-specific technologies in which the only source of user feedback is the military itself.

Consequently, the technical progress model in this simulation attempts to model both these features. To do so, a hybrid model was created. First, there is a baseline technology coefficient for each application area. Whenever a technology is fielded, the coefficient is multiplied by a learning factor (e.g., 1.1). This captures the learning from implementation. Second, there is an exponential growth model for each application area. This represents the learning from exogenous R&D activities. The two are multiplied together to determine the current technology level and are represented by the equation

$$Ce^{gt}$$

where C is the technology coefficient and g is the exogenous growth rate. Arriving technologies are assigned a performance as a random variation on this value. The parameters of this model can be adjusted to accommodate the specific situation of each application area. For example, technologies that are used commercially may have a high exogenous growth rate and low learning factor because their progress would continue regardless of military use. The reverse may be true for military-specific technologies since there would be little learning from commercial use.

4. Experimental Design

4.1 Simulation Parameters

As previously mentioned, the DoD has been relatively inconsistent in its implementation of its own policies, and evolutionary policies—in particular—are fairly new. Consequently, using historical data to set simulation parameters is particularly problematic. In fact, a RAND study to assess cost growth in weapon system programs found a number of issues in the available cost data for defense acquisition programs (Arena, Leonard, Murray & Younossi, 2006). Some of these issues include significant aggregation of data, baseline changes, changes in reporting guidelines, and incomplete data. The situation is worse for technology maturation. As indicated by the GAO, the DoD does not systematically track its technology development efforts (GAO, 2006, September). Furthermore, the introduction of TRL levels to the DoD is fairly recent, so there is little experience with their application in a DoD context. Since NASA has been using the TRL scale for some time, it would seemingly be a logical source of information regarding the cost and risk associated with maturing technologies through TRL levels. Unfortunately, a 2005 study at NASA to determine the cost and risk found that poor record keeping resulted in insufficient useful data to achieve statistically significant results (Kirn, 2005).

Fortunately, the aim of this study is not to precisely recreate the defense acquisition system as it is but to identify policy directions to determine how it should be. This, in combination with extensive sensitivity analysis, allows for a more reasonable margin of error in setting the simulation parameters. Consequently, the actual values used in the experiments are an amalgamation from several sources, including reports and studies from both government and commercial sources (Bodner & Rouse, 2007; DoD, 2007 April; DoD, 2007 August; Fox, 1988; GAO, 2007b; Kirn, 2005; Stevens & Burley, 1997).

4.2 Basic Experiment

In order to answer the research questions posed in this paper, three cases were developed. The three cases are variations on the key experimental variables, the Min TRL and the Fallback TRL. The Min TRL is the minimum maturity requirement for a technology used in an acquisition program, and the Fallback TRL is the minimum maturity selected when the first choice technology fails. The cases are as follows:

- **Base Case**—The base case most closely resembles the current modus operandi of the defense acquisition system. Technologies are selected at mid-TRL levels and final maturation occurs during the technology development phase of an acquisition effort. High performing, but immature technologies are preferred over more mature, proven technologies. If a technology fails, however, the program will fall back to a more mature technology.
  - Min TRL = 4
  - Fallback TRL = 7
- **Evolutionary Acquisition**—In this case, programs may only use fully mature technology. Maturation of technology is funded in the R&D system, and there is effectively no technology development phase. (Note that TRL 7 was chosen here because TRL levels 8 and 9 are system specific.)
  - Min TRL = 7
- **Revolutionary Acquisition**—Programs target maximum performance at all costs and, thus, always choose the technologies with the highest expected performance. When a technology fails, another top performer is selected in its place.
  - Min TRL = 4
  - Fallback TRL = 4

There are several outputs of interest. These are the cost of operating the entire acquisition system, the cost of an individual program, the annual capability growth rate, and the acquisition program length. Of course, we are interested in the long-run behavior of these outputs. Consequently, to perform the experiments, the simulation was run for a warm-up period in order to fully populate the technology portfolio, and then statistics were collected on the outputs of interest.

In particular, each simulation was run for a warm-up period of 50 years and then statistics were collected for another 150 years. There are 40 replications for each experimental case. As for the acquisition programs, there are three system types each requiring three technologies. Each of those technologies falls into one of six application areas. It was assumed that the three acquisition programs are homogenous in terms of cost and schedule risk, and it was also assumed that the application areas are homogeneous in terms of cost, schedule, and technical risk. The budget for the technology development process was set to $3 billion, and was allocated among the six stages so as to ensure a smooth flow of technologies through the system. It was also assumed that all of the stages are of equal length. This is simply to focus on the technical risk for the basic experiment. Finally, the technical progress model is identical for all six application areas and features a mix of exogenous technical progression and learning.

4.3 Sensitivity Analysis

The simulation developed is quite flexible and many different scenarios can be analyzed. A first order sensitivity analysis was performed, and the results are presented in Pennock (2008). It was found that the simulation outputs were particularly sensitive to five factors: the R&D budget size, the R&D budget distribution, the rate of technical learning, the technology development stage length, and production costs. The impact of the size of the R&D budget was examined by leaving the percent allocated per stage the same but varying the aggregate amount over the range of -50% to +50%. The budget distribution was analyzed by reducing the budget for stages 4, 5, and 6. This particular scenario was designed to represent the status quo of the defense technology development process. To understand the influence of the rate of technical learning, the learning factor from the technical progress model was varied between 1 (no learning) and 2. In the basic experiment, all technology development stages are one year in length. To understand the impact of stage length, the scenarios were run with stage lengths of two years and three years. Finally, the influence of production costs was analyzed by varying the cost rate from -100% to +100% of the baseline value.

5. Results and Analysis

5.1 Results of the Basic Experiment

First, we will consider the results of the basic experiment. The average values of each of the output statistics are displayed in Table 1. Note that for compactness, system specific outputs are only shown for system 1. The results are similar for the other two systems. The most obvious question is how do these program outputs compare to real acquisition programs? As far as program duration, the distributional parameters for concept development, system development, and production were derived from Fox (1988, p. 29). with an average program duration of 15 years. We see from Table 1 that the base case has an average duration of 14 years, which is fairly close. As for cost, Fox does not provide cost data, but a recent GAO report provides the cost and schedule performance of 62 current weapons system programs (GAO, 2007b). An analysis of these data reveals that the average program cost is approximately $16 billion. An important caveat is that these data cover a wide range of programs. Some are small upgrade programs that are short and inexpensive while others are major system of systems acquisitions that will take 30 years and cost hundreds of billions of dollars. Even so, we can see from Table 1 that the average program cost for the base case is approximately $16 billion. Thus, we can say that the simulation outputs are within the right order of magnitude for an “average” acquisition program.

Table 1. The Average Output Values over 40 Repetitions for the Scenarios of the Basic Experiment

Output	Base Case	Evolutionary	Revolutionary
Total Acquisition System Operating Cost ($ million, annualized)	5807	6410	5169
Capability Growth Rate (System 1)	0.16	0.179	0.138
Program Duration (System 1, years)	14.3	11.8	17.2
Program Cost (System 1, $ million)	16091	14668	16736

In order to understand these results fully, we will address each of the four outputs in turn. Figure 3a depicts the 95% confidence intervals for the average annual cost to operate the acquisition system. Clearly, evolutionary acquisition is the most expensive and revolutionary acquisition is the least expensive. If the technology policy is less aggressive with evolutionary acquisition, why would it be more expensive? To better understand this outcome, let us consider the average cost of the individual programs.

Figure 3b shows the confidence intervals for the average program cost to acquire a system of type 1. Here we see that the average program cost is actually lower with evolutionary acquisition than revolutionary acquisition. So as evolutionary acquisition supporters suggest, using mature technology must lower program cost. Then why does the acquisition system cost more to operate under evolutionary acquisition?

The answer is revealed when we examine the average program duration or cycle-time. In Figure 3c, we see that the program length is much shorter with evolutionary acquisition. With a shorter cycle-time, acquisitions happen more frequently. Each cycle imposes overhead costs including system development, production, and deployment costs. Since these overhead costs are far greater than any savings that would result from more efficient management of the technology portfolio, the overall cost rises.

But does the additional cost of evolutionary acquisition buy the DoD anything? Figure 3d reveals that evolutionary acquisition results in a superior annual capability growth rate. The annual capability growth rate is the “average” annual rate of capability improvement. Much like an interest rate, even small differences in the rate can result in a huge difference in the level of deployed capability over the long-run. Thus, we see that there is a cost/performance trade-off governed by the technology maturity requirement. Allowing less-mature technology hurts system performance because it takes longer to move technologies into the field, but since it incurs large production costs less often, it is also less expensive. Strictly enforcing maturity requirements, on the other hand, means shorter, less-expensive programs that achieve high performance by moving technologies into the field more quickly. Unfortunately, this incurs production costs more frequently and results in increased operating costs for the acquisition system as a whole.

In fact, by varying the technology policy we can move along a roughly linear frontier of cost/performance combinations. Figure 4 shows the cost and performance for all possible technology polices such that 1 ≤ Min TRL ≤ 7 and Fallback TRL ≥ Min TRL. At first, this result would seem to suggest that technology policy should not be strictly enforced as budgetary restrictions would force changes in technology policy to meet cost goals. Fortunately, this is not the case.

In order to maintain a consistent, evolutionary technology policy but retain the ability to trade performance for cost, all that is required is to insert a delay between acquisition cycles. Figure 5 depicts the cost/performance combinations for the evolutionary policy with inter-cycle delays ranging from 0 to 7 years. Also shown is the linear trend line from Figure 4. Clearly, the introduction of a delay allows the evolutionary policy to replicate the cost/performance combinations achieved through shifts in technology policy. Thus, for any given cost target, an efficient policy can be found by imposing the evolutionary maturity requirements in combination with the appropriate inter-acquisition cycle delay.

5.2 Sensitivity Results

The previous section presented the results of the basic experiment, but there remains a question of robustness. How stable are results? Are there any cases in which the evolutionary policy is not the best performing? While five scenarios were described in Section 4.3 above, due to space constraints only two will be presented here. For the remainder see Pennock (2008).

First, we consider the distribution of the R&D budget among the stages. In particular, this scenario is designed to represent a situation that is often referred to as crossing the chasm. Crossing the chasm describes the difficulty that technology development efforts often encounter in moving through the middle stages of technology maturation because of a scarcity of funding. To simulate this scenario, funding for stages 4, 5, and 6 was varied over a range of 25% to 100% of the baseline value. Figure 6 reveals that the best policy from a performance standpoint is quite sensitive to the level of middle-stage funding.

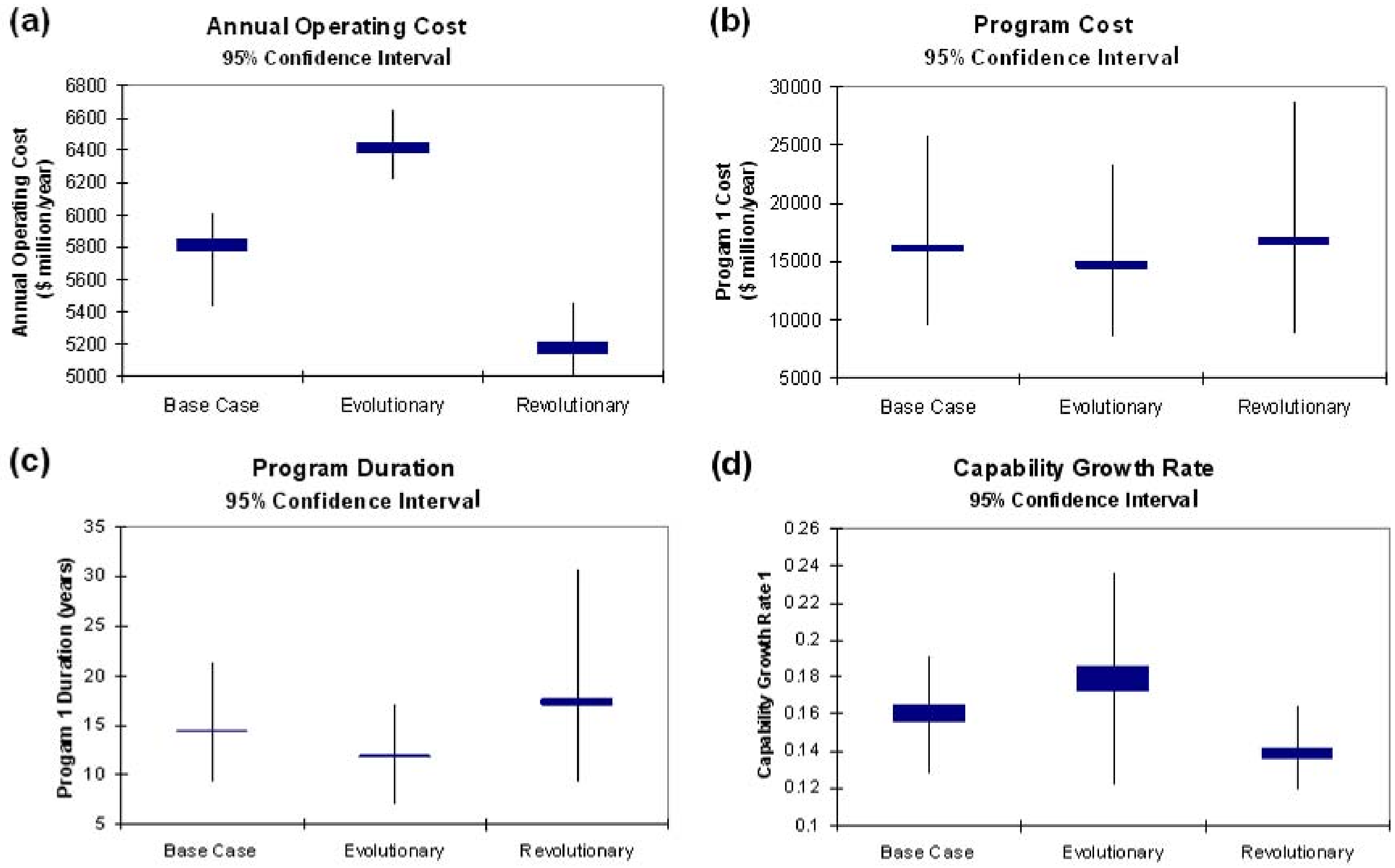


Figure 3. 95% Confidence Intervals by Scenario for the Mean Values of Each of the Four Primary Outputs

Note: The height of the box represents the width of the confidence interval, and the vertical lines represent the range of realized values.

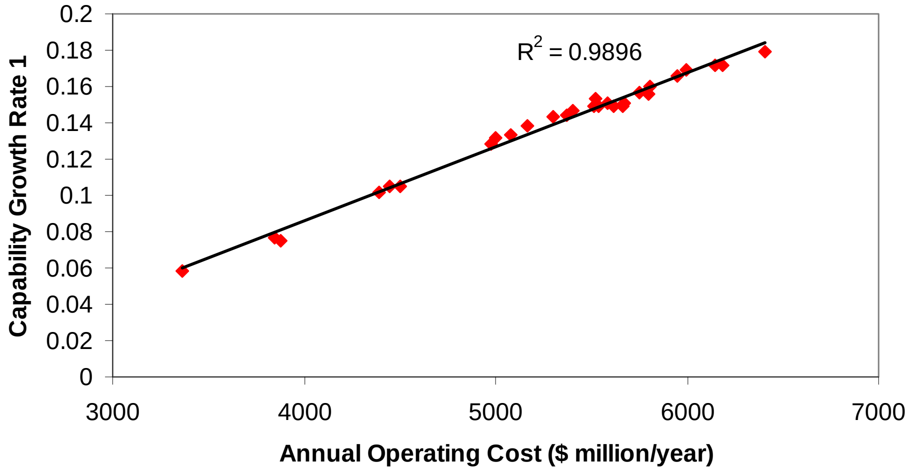


Figure 4. Cost/Performance Trade-off for All Possible Technology Policies with a Linear Trend Line

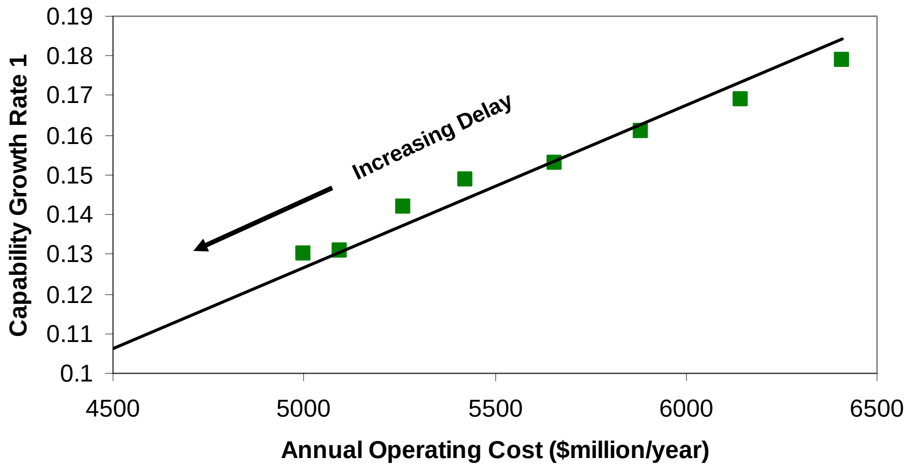


Figure 5. Cost/Performance Trade-off Replicated through the Evolutionary Policy with an Inter-cycle Delay

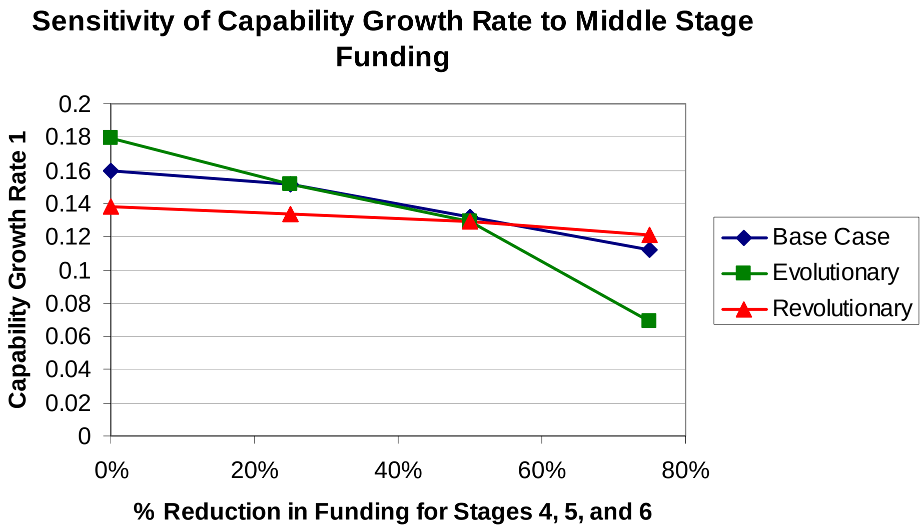


Figure 6. Capability Growth Rate when Middle-stage R&D Funding Is Cut

As we would expect, the evolutionary policy is the most sensitive since it is dependent upon a constant supply of mature technologies. On the opposite end, the revolutionary policy is the most robust since it can provide its own middle-stage funding, and once again, the base case falls in between. Given the varied rates of performance decay among the three policies, there are domains in which each is dominant. When R&D is well funded, the evolutionary policy provides superior performance. As middle-stage funding is reduced by more than 25%, the performance of the base case policy begins to exceed the performance of the evolutionary policy. As funding declines further, the revolutionary policy becomes the top performing policy.

Of all of the scenarios presented in this paper, the crossing the chasm scenario is probably the most similar to business as usual at the DoD. Typically, S&T funding covers early-stage technology development, but once technologies reach the middle stages, the only readily available source of funding is through an acquisition effort. The base case policy is also fairly similar to the risk mitigation strategy that many acquisition programs use: try to utilize the most promising technology, but if that fails, fall back to the existing, mature technology. Thus, it would seem that given the circumstances that most acquisition programs operate under, the business as usual policy is quite rational. Of course, it should be pointed out that all of the acquisition policies perform better when middle-stage R&D is well funded.

The final scenario represents the impact of production costs on the affordability of evolutionary acquisition. The production cost rate was varied from zero to $8 billion per year. Figure 7 reveals that as procurement cost increases, the spread between the operating costs of the three policies increases. The shorter the acquisition cycle, the more frequently production costs are incurred and, consequently, the greater the impact of an increase in production costs. Conversely, the lower production costs are, the more cost effective evolutionary acquisition becomes.

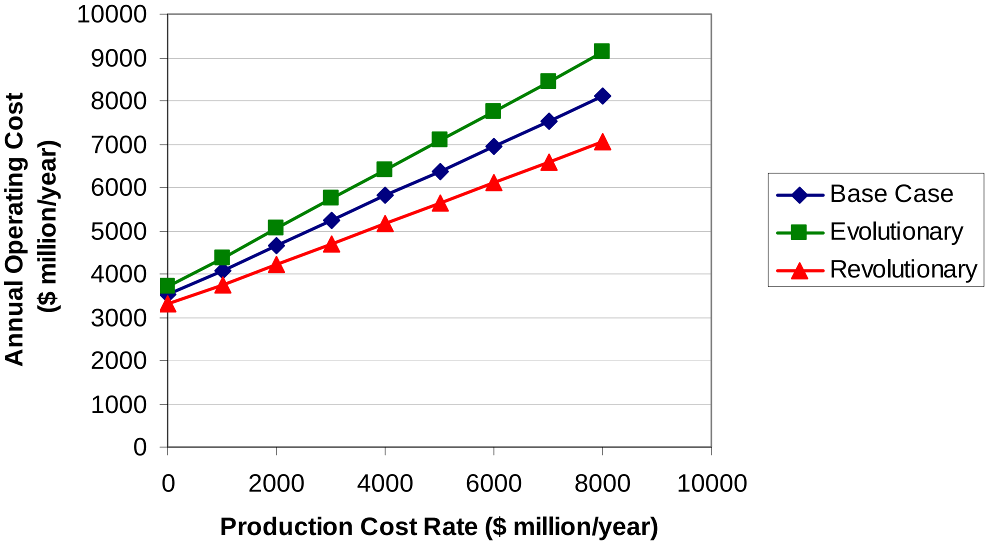


Figure 7. The Annual Acquisition System Operating Cost as a Function of the Production Cost Rate

6. Discussion

The production cost scenario raises several issues regarding evolutionary acquisition. Clearly, the more expensive it is to produce and deploy the next iteration of a system, the less affordable evolutionary acquisition becomes. But, of course, that is dependent upon the nature of the system under consideration, and this is a key difference between evolutionary practices in a commercial setting versus a defense setting. A commercial firm does not purchase its own product. In fact, if we take the example of a car manufacturer—there is always a substantial portion of the customer base that is looking to buy a new car. Thus, the car manufacturer is going to build and sell cars continuously. The costs of upgrading a model might include the costs of any technology development, the cost of changing the design, and the cost of any retooling that must be done at production facilities. If the manufacturer is particularly successful, it may gain market share from its competitors, and thus, the investment pays for itself. Consequently, a commercial firm can actually make more money from cycling faster and using an evolutionary approach. When the DoD would like to buy a new weapon system, it must pay for all of the same development costs and purchase the product. Furthermore, if through more rapid acquisition cycles the DoD improves the performance growth rate of its systems, it may outperform its adversaries, but it does not generate a monetary return to help fund the faster pace of system development.

Thus, the cost of evolutionary acquisition is critically dependent upon the length and cost of stages in the system acquisition lifecycle. The simulation model presented in this paper was generic in the sense that it assumed that something was acquired in each cycle, but it did not differentiate between a new system design or a product upgrade. Representing either case could be achieved by simply changing the cost and duration parameters in the model. The key outcome of the evolutionary policy was that simply employing mature technology shortened the acquisition cycle and reduced the cost of each cycle. In the examples above, however, the decline in cycle costs from more efficient technology development alone was not sufficient to compensate for the increase in the cycle rate. Thus, total acquisition costs rose with evolutionary acquisition. Some have suggested, however, that the length and cost of other phases of the acquisition lifecycle would decline under evolutionary acquisition as well. The idea is that if acquisition programs are less ambitious and shorter, development will be easier and there will be fewer problems with unstable funding. Thus, we should expect lower system development and procurement costs as well. Consequently, the question becomes, if the costs of system development and production decline under evolutionary acquisition, does evolutionary acquisition then become less expensive than more traditional methods?

To consider this question let us develop a very simple model for the cost of operating the defense acquisition system. First, we define the following symbols:

- r_{ij} ≡ the acquisition cycle rate for system *i* under policy *j* in cycles per year.
- C_{ij} ≡ the cost per acquisition cycle for system *i* under policy *j*.
- K_j ≡ the total cost per year for operating the defense R&D system under policy *j*.
- A_j ≡ the annual cost of operating the defense acquisition system under policy *j*.

We can define the cost rate to operate the acquisition system under policy *j* as

$$A_j = \sum_{i=1}^{n} r_{ij} C_{ij} + K_j$$

where *n* is the number of systems begin acquired. Thus, if policy *e* represents evolutionary acquisition and policy *t* represents traditional acquisition, then evolutionary acquisition would be less expensive if $A_e < A_t$. For the moment, let us assume that all systems being acquired have identical cost and cycle rates. This leaves us with the relationship

$$n r_e C_e + K_e < n r_t C_t + K_t.$$

Furthermore, if we assume that we keep our R&D budget fixed we can simplify even further to yield

$$\frac{C_e}{C_t} < \frac{r_t}{r_e}.$$

Of course, since the rate of acquisition is slower under the traditional acquisition policy, the right-hand side will be strictly less than one. This implies that a simple decline in program costs from evolutionary acquisition is not sufficient to reduce the total cost to operate the acquisition system. Instead, program costs must decline sufficiently to offset the increase in the rate of acquisition.

To better illustrate this point, imagine that acquisition cycles were weekly and cost $10. The operating cost would be $10 per week. Now let us assume that we institute a new policy that reduces cycle costs to $8 per cycle, but the cycles now occur twice as fast. That means that under the new policy, the operating cost would be $16 per week. Thus, even though the cost per cycle decreased, the total cost increased.

When we consider defense acquisition cycles, if the system development and procurement costs also drop under evolutionary acquisition, it might seem to suggest that we could overcome this deficit. If, however, the durations of system development and technology development also decrease, then the equivalent cost threshold becomes even more difficult to reach. Furthermore, if we consider spiral development when there are several short, overlapping cycles, we see that we would require fairly low development, production, and deployment costs to compensate for the speed of the cycles.

Thus, the critical question becomes, how does evolutionary acquisition affect the length and cost of development and procurement activities versus a traditional single-step to capability approach? This is not a trivial question, and the answer will likely depend on the type of system being acquired. Upgrades to complex, integrated systems can lead to substantial design modifications to accommodate even seemingly simple changes and using more mature technologies does not correlate to easier integration (Smaling & de Weck, 2007). In fact, experiences at Westland Helicopters indicate that even when a system such as a military helicopter is designed with modularity and upgradeability in mind, changes can unexpectedly propagate through large portions of the system design (Clarkson, Simons & Eckert, 2004; Eckert, Clarkson & Zanker, 2004). At the other end of the spectrum, systems with very loose coupling between system components may be quite amenable to rapid upgrade and change. Perhaps the most extreme example of this type of system is the Internet, in which the system architecture changes continuously without any supervision or control.

Thus, this issue merits substantial additional research and is really the determining factor regarding evolutionary acquisition's potential for cost savings. This is not to suggest that if the costs of acquiring a particular system type do not decline under evolutionary acquisition that the approach is useless. The results of this study suggest that evolutionary acquisition delivers other benefits such as a boost in the capability of systems actually deployed in the field. Instead, it simply means that additional capability will continue to come at additional cost. Consequently, cost and performance may be traded off by simply, appropriately spacing acquisition cycles.

7. Conclusions and Further Research

The results from this simulation study lead to some highly suggestive findings and critical avenues for future research. First and foremost, with a first-order representation of the acquisition system, the results suggest that the adoption of evolutionary acquisition policies has the potential to improve the performance of deployed systems. However, lower operating costs for the defense acquisition system are not automatic. While each individual program should be less expensive under evolutionary acquisition policies, the faster acquisition cycle-time means that development, production, and deployment costs are incurred more frequently. This may overwhelm any cost savings from managing technology development more efficiently. As discussed in Section 6, these cycle costs must decline sufficiently under evolutionary acquisition to achieve net cost savings. Thus, depending on the type of system being acquired, evolutionary acquisition may actually be more expensive than traditional means of acquiring military systems. This is a critical issue for future research. However, this should not be interpreted as an endorsement of traditional acquisition methods. Instead, acquisition cycle-time can be used to control the costs of an evolutionary policy without reverting to a traditional approach that employs immature technology. A requirement for mature technologies can be consistently imposed with the next acquisition cycle beginning only when it is affordable.

There are some important caveats on this conclusion, however. First, the above results are more significant for military-specific technologies than commercial technologies. Commercial technologies will continue to develop and improve regardless of the actions of the DoD because the DoD is actually a small player in the market. One example is microprocessor technology. On the Comanche helicopter program, the mission processing technology was changed three times because Intel introduced newer processor models faster than the DoD could develop an advanced combat helicopter (Rogers & Birmingham, 2004). For military-specific technologies, however, forward progress is dependent upon actually testing and fielding a technology and gathering user feedback. Thus, the faster acquisition cycles are, the faster learning can be incorporated into new technologies under development. Of course, faster acquisition cycles also mean that exogenously developed commercial technologies can also be moved into the field faster.

Second, evolutionary acquisition policies do not function well when the R&D process is underfunded. Evolutionary acquisition depends on a steady stream of mature technologies. When the research pipeline is "starved," not only does the performance of deployed systems decline on average, but it also becomes more unpredictable. More traditional acquisition methods mitigate this risk by using an acquisition effort to secure funding for technology development.

Third, the underfunding of middle-stage technologies, as is typical for government technology development (Cornford & Sarsfield, 2004), also adversely impacts evolutionary acquisition policies. Under these circumstances, traditional approaches to acquisition are actually superior to evolutionary methods since they mitigate the risk of technologies failing to cross the chasm. Thus, it would seem that business as usual is quite reasonable under the current funding environment for military R&D activities. Though, it is important to point out that traditional acquisition policies under this scenario still underperform evolutionary policies when R&D is fully funded.

Finally, there are several features of the current defense acquisition system that were not considered in this analysis. First and foremost among these is concurrency. For major acquisition efforts there is often substantial overlap between the technology development, system development, and production phases. While this is often an attempt to compress an otherwise long acquisition cycle, the resulting rework often increases costs and leads to performance shortfalls. This problem has been extensively documented elsewhere, and there is no need for it to be recapitulated here. If, however, the imposition of evolutionary acquisition and its shorter acquisition cycles reduced the temptation to use a concurrent acquisition strategy, it is possible that there could be a net cost savings through the reduction of rework, but that determination must be relegated to future work. Other features of defense acquisition not considered in this model are operations and maintenance costs, basic research funding, non-centralized acquisition management, program cancellation, program budgeting, the capacity of the industrial base, the capacity of the government to consume, and system integration issues. Each of these factors certainly influences the behavior and cost effectiveness of the defense acquisition system and may be examined in future work.

What we can ultimately derive from this study is that, at least to a first order, there are definite benefits to the better management and development of new technologies implied in evolutionary acquisition. A well-managed technology portfolio leads to the development of technology options, which creates the flexibility to maximize the ability of acquired systems to meet emerging threats. Traditional programs, through their early commitment to particular approaches and technologies, sacrifice some of this flexibility. The outstanding question raised is whether the increased flexibility created by evolutionary acquisition comes at additional cost. What this study revealed is that net cost savings are not automatic. Additional research is required to determine under what circumstances they are possible.

List of References

Arena, M.V., Leonard, R.S., Murray, S.E., & Younossi, O. (2006). *Historical cost growth of completed weapon system programs* (TR-343-AF). Santa Monica, CA: RAND.

Bodner, D.A., & Rouse, W.B. (2007). Understanding R&D value creation with organizational simulation. *Systems Engineering, 10*(1), 64-82.

Clarkson, P.J., Simons, C., & Eckert, C. (2004). Predicting change propagation in complex design. *Transactions of the ASME*, *126*, 788-797.

Cornford, S.L., & Sarsfield, L. (2004). Quantitative methods for maturing and infusing advanced spacecraft technology. *IEEE Aerospace Conference Proceedings,* 663-681. Los Alamitos: IEEE.

DoD. (2006*). Defense acquisition guidebook*. Retrieved November 12, 2007, from http://akss.dau.mil/dag/welcome.asp

DoD. (2007, April 6). *Selected acquisition report (SAR) summary tables*, Washington, DC: Author.

DoD. (2007, August 14). *Selected acquisition report (SAR) summary tables.* Washington, DC: Author.

Eckert, C., Clarkson, P.J., & Zanker, W. (2004). Change and customization in complex engineering domains. *Research in Engineering Design*, *15*, 1-21.

Fox, J.R. (1988). *The defense management challenge: Weapons acquisition*. Boston: Harvard Business School Press.

GAO. (2003, November 10). *Defense acquisitions: DoD's revised policy emphasizes best practices, but more controls are needed* (GAO-04-53) Washington, DC: Author.

GAO. (2006, April 13). *Defense acquisitions: Major weapon systems continue to experience cost and schedule problems under DoD's revised policy* (GAO-06-368). Washington, DC: Author.

GAO. (2006, September 5). *Best practices: Stronger practices needed to improve DoD technology transition processes* (GAO-06-883). Washington, DC: Author.

GAO. (2007a, March 30). *Best practices: An integrated portfolio management approach to weapon system investments could improve DoD's acquisition outcomes* (GAO-07-388). Washington, DC: Author.

GAO. (2007b, March 30). *Defense acquisitions: Assessments of selected weapons programs* (GAO-07-406SP). Washington, DC: Author.

Kirn, M. (2005, August). *Examining NASA's use of technology readiness levels*. Techical report. Washington, DC: NASA,.

Lorell, M.A., Lowell, J.F., & Younossi, O. (2006*). Evolutionary acquisition: Implementation challenges for defense space programs*. Santa Monica, CA: RAND.

NRC. (2008). *Pre-Milestone A and early-phase systems engineering: A retrospective review and benefits for future air forces systems acquisition*. Washington, DC: The National Academies Press.

Pennock, M.J. (2008). *A systematic analysis of the cost and performance impact of acquisition technology policy.* Working paper, Tennenbaum Institute. Atlanta, GA: Georgia Institute of Technology.

Smaling, R., & de Weck, O. (2007). Assessing risks and opportunities of technology infusion in system design. *Systems Engineering*, *10*(1), 1-25.

Rogers, E.W. ,& Birmingham, R.P. (2004, January-April). A ten-year review of the vision]for transforming the defense acquisition system. *Defense Acquisition Review Journal*, 37-61.

Stevens, G.A., & Burley, J. (1997). 3,000 raw ideas = 1 commercial success! *Research Technology Management*, *40*(3), 16-27.

System Maturity Indices for Decision Support in the Defense Acquisition Process

Presenter: Brian J. Sauser received his BS in Agriculture Development from Texas A&M University, MS in Bioresource Engineering from Rutgers, The State University of New Jersey, and PhD in Technology Management from Stevens Institute of Technology. He is currently an Assistant Professor at Stevens Institute of Technology in the School of Systems and Enterprises.

Brian J. Sauser
Stevens Institute of Technology
School of Systems and Enterprises
One Castle Point on Hudson
Hoboken, NJ 07030
Tel: (201) 216-8589
Fax: (201) 216-5541
E-mail: bsauser@stevens.edu

Presenter: Jose E. Ramirez-Marquez received his BS in Actuarial Science from Universidad Nacional Autónoma de México, MS in Statistics and MS and PhD in Industrial and Systems Engineering from Rutgers, The State University of New Jersey. He is currently an Assistant Professor at Stevens Institute of Technology in the School of Systems and Enterprises.

Jose E. Ramirez-Marquez
Stevens Institute of Technology
School of Systems and Enterprises
One Castle Point on Hudson
Hoboken, NJ 07030
Tel: (201) 216-8003
Fax: (201) 216-5541
jmarquez@stevens.edu

Author: Romulo B. Magnaye received his BS in Mining Engineering and MBA from the University of the Philippines and a postgraduate research Diploma in Mineral Technology from Camborne School of Mines in England. He is currently pursuing his PhD in Engineering Management at Stevens Institute of Technology in the School of Systems and Enterprises.

Romulo B. Magnaye
Stevens Institute of Technology
School of Systems and Enterprises
One Castle Point on Hudson
Hoboken, NJ 07030
Fax: (201) 216-5541
rmagnaye@stevens.edu

Author: Weiping Tan received his BS in Automation from Beijing Institute of Technology and is currently pursuing his PhD in Engineering Management at Stevens Institute of Technology in the School of Systems and Enterprises.

Weiping Tan
Stevens Institute of Technology
School of Systems and Enterprises
One Castle Point on Hudson
Hoboken, NJ 07030
Fax: (201) 216-5541
wtan@stevens.edu

Abstract

The Technology Readiness Level (TRL) scale is a measure of maturity of an individual technology, with a view towards operational use in a system context. A comprehensive set of concerns becomes relevant when this metric is abstracted from an individual technology to a system context, which may involve interplay among multiple technologies that are integrated through the defense acquisition process. This paper proposes the development of a system-focused approach for managing system development and making effective and efficient decisions during the defense acquisition process. For this to be accomplished, a new System Readiness Level (SRL) index will incorporate both the current TRL scale and the concept of an integration readiness level (IRL). This paper describes the foundations for the SRL and provides techniques for determining current and future readiness of a system to determine its position in the defense acquisition process. In addition, it proposes optimization models than can provide management with an optimal development plan that can meet the objectives of the development team, based on constrained resources. These, in turn, can become the foundation for the development of a monitoring and evaluation tool that will be analogous to Earned Value Management used in project management.

1. Introduction

In theory, technology and system development follow similar evolution (or maturation) paths; a technology is inserted into a system (e.g., spiral development) based on its maturity, functionality and environmental readiness, and ability to interoperate with the intended system. However, the assessments made during the acquisition lifecycle that support these decisions are not always effective, efficient, or well developed. Recently, the Government Accounting Office (GAO) stated that many of the programs in the Department of Defense (DoD) plan to hold design reviews or to make a production decision without demonstrating the level of technology maturity that should have been there before the start of development (GAO, 1999). In many US government agencies and contractors, Technology Readiness Level (TRL) is used to assess the maturity of evolving technologies (materials, components, devices, etc.) prior to incorporating a technology into a system or subsystem. In the 1990s the National Aeronautics and Space Administration (NASA) instituted this nine-level metric as a systematic metric/measurement approach to assess the maturity of a particular technology and to allow consistent comparison of maturity between different types of technologies (Mankins, 2002). Given the pragmatic benefits of this concept, in 1999, the DoD embraced a similar TRL concept (USD(AT&L), 2005: DoD, 2005). While the use of TRL is similar in these organizations, TRL was not intended to measure the integration of technologies, but was to be used as ontology for contracting support (Sadin, Povinelli, & Rosen, 1989), thus TRL does not address:

- A complete representation of the (difficulty of) integration of the subject technology or subsystems into an operational system (Mankins, 2002; Dowling & Pardoe, 2005; Valerdi & Kohl, 2004),
- The uncertainty that may be expected in moving through the maturation of TRL (Mankins, 2002; Dowling & Pardoe, 2005; Smith, 2005; Cundiff, 2002), and
- Comparative analysis techniques for alternative TRLs (Mankins, 2002; Dowling & Pardoe, 2005; Smith, 2005; Cundiff, 2002).

Based on these fundamental conjectures, a more comprehensive set of concerns becomes relevant when TRL is abstracted from the level of an individual technology to a system context, which usually involves the interplay among multiple technologies. Similarly relevant is the case in which these technologies are integrated through the defense acquisition process. That is, component level considerations relating to integration, interoperability, and sustainment become equally or more important from a systems perspective in an operational environment.

Technology insertion as part of a defense acquisition process needs a quantitative assessment tool that can determine whether a group of separate technology components with their associated (and demonstrated) TRL ratings can be integrated into a larger complex system at a reasonably low risk in order to perform a required function or mission at some performance level.

However, before such a tool can be developed, we must first address the issue of measuring the maturity of the integration elements. The very first attempt to address this was done by Mankins (2002) when he proposed an Integrated Technology Analysis Methodology to estimate an Integrated Technology Index (ITI). The ITI was then used for a comparative ranking of competing advanced systems. The study brought to the forefront the difficulty of progressing through the TRL index and choosing between competing alternative technologies; it did not adequately address the integration aspects of systems development.

Based on concerns for successful insertion of technologies into a system, the Ministry of Defence in the United Kingdom developed a Technology Insertion Metric that includes, among other things an Integration Maturity Level (Dowling & Pardoe, 2005). Building upon these efforts, Gove, Sauser, and Ramirez-Marquez (2008) performed a thorough review of aerospace and defense-related literature to identify the requirements for developing a seven-level integration metric which they called Integration Readiness Level (IRL). It has since evolved into the nine-level concept (Gove, 2007) described in Table 1 below.

Table 1. Integration Readiness Levels

IRL	Definition
9	Integration is **Mission Proven** through successful mission operations.
8	Actual integration is completed and **Mission Qualified** through test and demonstration, in the system environment.
7	The integration of technologies has been **Verified and Validated** with sufficient detail to be actionable.
6	The integrating technologies can **Accept, Translate, and Structure Information** for its intended application.
5	There is sufficient **Control** between technologies necessary to establish, manage, and terminate the integration.
4	There is sufficient detail in the **Quality and Assurance** of the integration between technologies.
3	There is **Compatibility** (i.e., common language) between technologies to orderly and efficiently integrate and interact.

2	There is some level of specificity to characterize the I**nteraction** (i.e., ability to influence) between technologies through their interface.
1	An **Interface** between technologies has been identified with sufficient detail to allow characterization of the relationship.

IRL is a systematic measurement of the interfacing of compatible interactions for various technologies and the consistent comparison of the maturity between integration points. The introduction of IRL to the assessment process not only checks the place of technology on an integration readiness scale, but also presents a direction for improving integration with other technologies. Just as TRL has been used to assess the risk associated with developing technologies, IRL is designed to assess the risk associated with integrating these technologies. Now that both the technologies and integration elements can be assessed and mapped along an objective numerical scale, the next challenge is to develop a metric that can assess the maturity of the entire system that is under development. Sauser, Ramirez-Marquez, Henry, and DiMarzio (2008) were able to demonstrate how using a normalized matrix of pair-wise comparisons of TRLs and IRLs for any system under development can yield a measure of system maturity, called Systems Readiness Level (SRL). The SRL metric can be used to determine the maturity of a system and its status within a developmental lifecycle. Table 2 presents the definitions of the various levels of the SRL and a representation of how the SRL index correlates to a systems engineering lifecycle.

Table 2. System Readiness Levels

SRL	Acquisition Phase	Definitions
0.90 to 1.00	*Operations & Support*	Execute a support program that meets operational support performance requirements and sustains the system in the most cost-effective manner over its total lifecycle.
0.70 to 0.89	*Production*	Achieve operational capability that satisfies mission needs.
0.60 to 0.79	*System Development & Demonstration*	Develop system capability or (increments thereof); reduce integration and manufacturing risk; ensure operational supportability; reduce logistics footprint; implement human systems integration; design for production; ensure affordability and protection of critical program information; and demonstrate system integration, interoperability, safety and utility.
0.40 to 0.59	*Technology Development*	Reduce technology risks and determine appropriate set of technologies to integrate into a full system.
0.10 to 0.39	*Concept Refinement*	Refine initial concept; develop system/technology strategy.

NOTE: These ranges have been derived conceptually and are undergoing field verification and validation under Naval Postgraduate School Contract # N00244-08-0005.

2. Calculating System Readiness Level

The computation of the SRL is a function of two matrices:

1. Matrix **TRL** provides a blueprint of the state of the system with respect to the readiness of its technologies. That is, **TRL** is defined as a vector with *n* entries for which the *i*th entry defines the TRL of the *i*th technology.
2. Matrix **IRL** illustrates how the different technologies are integrated from a system perspective. **IRL** defined as an *n*×*n* matrix for which the element IRL$_{ij}$ represents the maturity of integration between the *i*th and *j*th technologies.

In these matrices, the standard TRL and IRL levels corresponding to values from 1 through 9 should be normalized. Also, it has been assumed that on the one hand, a value of 0 for element IRL$_{ij}$ defines that the *i*th and *j*th technologies are impossible to integrate. On the other hand, a value of 1 for element IRL$_{ij}$ can be understood as one of the following, with respect to the *i*th and *j*th technologies: 1) are completely compatible within the total system, 2) do not interfere with each others functions, 3) require no modification of the individual technologies, and 4) require no integration linkage development. Also, it is important to note that IRL$_{ii}$ may have a value lower than 1, illustrating that the technology may be a composite of different sub-technologies that are not absolutely mature.

In any system, each of the constituent technologies is connected to a minimum of one other technology through a bi-directional integration. How each technology is integrated with other technologies is used to formulate an equation for calculating SRL that is a function of the TRL and IRL values of the technologies and the interactions that form the system. In order to estimate a value of SRL from the TRL and IRL values, we propose a normalized matrix of pair-wise comparison of TRL and IRL indices. That is, for a system with *n* technologies, we first formulate a TRL matrix, labeled [TRL]. This matrix is a single column matrix containing the values of the TRL of each technology in the system. In this respect, [TRL] is defined in Equation 1, where TRL$_i$ is the TRL of technology *i*.

Equation 1.

$$[TRL]_{n\times 1} = \begin{bmatrix} TRL_1 \\ TRL_2 \\ \ldots \\ TRL_n \end{bmatrix}$$

Second, an IRL matrix is created as a symmetric square matrix (of size *n*×*n*) of all possible integrations between any two technologies in the system. For a system with *n* technologies, [IRL] is defined in Equation 2, where IRL$_{ij}$ is the IRL between technologies *i* and *j*. It is important to note that whenever two technologies are not planned for integration, the IRL value assumed for these specific technologies is the hypothetical integration of a technology *i* to itself; therefore, it is given the maximum level of 9 and is denoted by IRL$_i$

Equation 2.

$$[IRL]_{n\times n} = \begin{bmatrix} IRL_{11} & IRL_{12} & \ldots & IRL_{1n} \\ IRL_{21} & IRL_{22} & \ldots & IRL_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ IRL_{n1} & IRL_{n2} & \ldots & IRL_{nn} \end{bmatrix}$$

Although the original values for both TRL and IRL can be used, the use of normalized values allows a more accurate comparison when comparing the use of competing technologies. Thus, the values used in [TRL] and [IRL] are normalized (0,1) from the original (1,9) levels. Based on these two matrices, an SRL matrix is obtained by obtaining the product of the TRL and IRL matrices, as shown in Equation 3.

Equation 3. $$[SRL]_{n\times 1} = [IRL]_{n\times n} \times [TRL]_{n\times 1}$$

The SRL matrix consists of one element for each of the constituent technologies. From an integration perspective, it quantifies the readiness level of a specific technology with respect to every other technology in the system while also accounting for the development state of each technology through TRL. Mathematically, for a system with *n* technologies, [SRL] is as shown in Equation 4.

Equation 4. $$[SRL] = \begin{bmatrix} SRL_1 \\ SRL_2 \\ \ldots \\ SRL_n \end{bmatrix} = \begin{bmatrix} IRL_{11}TRL_1 + IRL_{12}TRL_2 + \ldots + IRL_{1n}TRL_n \\ IRL_{21}TRL_1 + IRL_{22}TRL_2 + \ldots + IRL_{2n}TRL_n \\ \ldots \\ IRL_{n1}TRL_1 + IRL_{n2}TRL_2 + \ldots + IRL_{nn}TRL_n \end{bmatrix}$$

where $IRL_{ij}=IRL_{ji}$.

Each of the SRL values obtained in Equation 4 would fall within the interval (0,n). For consistency, these values of SRL should be divided by n to obtain the normalized value between (0,1). Notice that [SRL] can be used as a decision-making tool since its elements provide a prioritization guide of the system's technologies and integrations. Thus, [SRL] can point out deficiencies in the maturation process.

The SRL for the complete system is the average of all such normalized SRL values, as shown in Equation 5. Equal weights are given to each technology, and hence, a simple average is estimated. A standard deviation can also be calculated to indicate the variation in the system maturity and parity in subsystem development.

Equation 5. $$SRL = \frac{\left(\frac{SRL_1}{n_1} + \frac{SRL_2}{n_2} + \ldots + \frac{SRL_n}{n_n} \right)}{n}$$

where n_i is the number of integrations with technology *i*.

3. An Example of SRL Calculation

The following example will use a real blue-water ship that is currently under development to show the steps involved in calculating its SRL value. This system example will be referred to as *System X* and its contemplated architecture is shown in Figure 1. For this system, the following matrices can be created for the TRL and IRL, based on the definitions presented earlier in Tables 1 and 2.

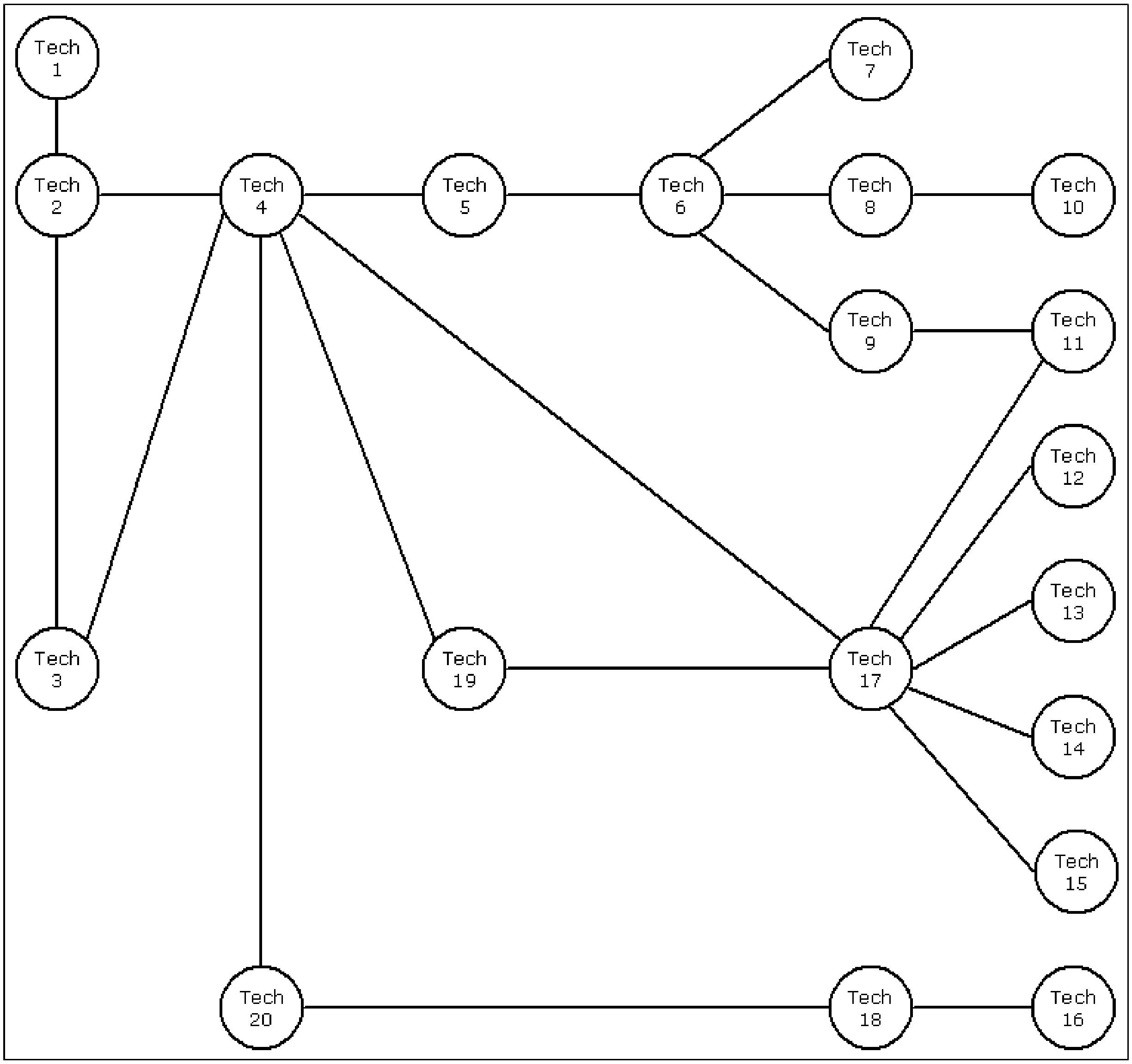


Figure 1. Schematic Architecture of System X

Equation 1a.

$$[TRL]_{20\times1} = \begin{bmatrix} TRL_1 \\ TRL_2 \\ \ldots \\ TRL_{20} \end{bmatrix} = [9 \quad 9 \quad 9 \quad 7 \quad 6 \quad 9 \quad 9 \quad 7 \quad 6 \quad 9 \quad 9 \quad 8 \quad 7 \quad 6 \quad 8 \quad 7 \quad 6 \quad 8 \quad 9 \quad 9]^T$$

Equation 2a.

$$[IRL]_{20\times 20}=\begin{bmatrix} IRL_{11} & IRL_{12} & \dots & IRL_{1n} \\ IRL_{21} & IRL_{22} & \dots & IRL_{2n} \\ \dots & \dots & \dots & \dots \\ IRL_{n1} & IRL_{n2} & \dots & IRL_{nn} \end{bmatrix}$$

$$=\begin{bmatrix}
9 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & 9 & 9 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 9 & 9 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 8 & 7 & 9 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 5 & 6 \\
0 & 0 & 0 & 6 & 9 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 7 & 9 & 9 & 8 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 9 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 8 & 0 & 9 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 9 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 0 & 9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 9 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 8 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 8 & 0 & 0 \\
0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 9 & 9 & 9 & 8 & 0 & 9 & 0 & 9 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 9 & 0 & 7 \\
0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 9 & 0 \\
0 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 9
\end{bmatrix}$$

As indicated in the above integration matrix, we assign an IRL value of 0 when there is no integration link contemplated between any 2 technologies. For integration to itself, an IRL value of 9 is used. After normalization of the [TRL] and [IRL] matrices, calculate [SRL] as follows:

Equations 3a and 4a.

$$[SRL]=[IRL]_{20\times 20}[TRL]_{20\times 1}=\begin{bmatrix} SRL_1 \\ SRL_2 \\ \dots \\ SRL_{20} \end{bmatrix}$$

Table 3. Individual SRL Values

SRL_1	SRL_2	SRL_3	SRL_4	SRL_5	SRL_6	SRL_7	SRL_8	SRL_9	SRL_{10}
2.000	3.691	2.605	4.481	1.963	3.728	2.000	2.333	2.000	1.519
SRL_{11}	SRL_{12}	SRL_{13}	SRL_{14}	SRL_{15}	SRL_{16}	SRL_{17}	SRL_{18}	SRL_{19}	SRL_{20}
1.556	1.444	1.333	1.481	1.568	5.778	2.358	2.099	2.210	1.519

Equation 5a.

$$Composite \quad SRL=\frac{\left(\frac{SRL_1}{n_1}+\frac{SRL_2}{n_2}+\dots+\frac{SRL_n}{n_n}\right)}{n}=\frac{\left(\frac{SRL_1}{2}+\frac{SRL_2}{4}+\dots+\frac{SRL_{20}}{3}\right)}{20}=0.763$$

The calculated Composite SRL index indicates that the system under development is currently in the System Development and Demonstration phase. Aside from providing an assessment of overall system development, it can also be a guide in prioritizing potential areas that require further development. This new index can then interact with decision-making tools for the potential acquisition of systems, which involve the dependency and interplay among performance, availability (reliability, maintainability, and supportability), process efficiency (system operations, maintenance, and logistics support), system lifecycle cost, and system maturity (measured by SRL). The overarching perspective of this methodology provides a context for the "trade space" available to a systems engineer or program manager, along with the articulation of the overall objective of maximizing the operational effectiveness of systems.

4. Potential Applications of SRL: Future Research

Given the ability to estimate readiness of a system under development, organizations can systematically evaluate the implications of using alternative technologies or system architectures, prepare development plans that optimize the objectives of the development team, and eventually be able to evaluate and monitor the progress of the development effort to identify problem areas and corrective measures.

4.1 Optimization Models

In the defense acquisition process, there are factors that may strategically alter the decision to develop one system over another; supersede a new, more functional system over another; determine if a system or technology has become inadequate due to changes in other systems or technologies; invest in the development of a new system or maintain existing systems; and classify a systems obsolescence and longevity. To address these challenges, we can use SRL as a method for determining current and future readiness of a system in order to determine its position in the defense acquisition process. While identifying the current SRL of a system can provide managerial insight, optimizing the future value of this index based on constrained resources will enhance managerial capabilities.

The optimization of SRL based on resource allocation can allow for decisions to be made regarding the trade-offs among: 1) system attributes such as availability, performance, efficiency, and total ownership cost and 2) the components necessary for producing affordable system operational effectiveness (pp. 14-15). These attributes have objectives and ranges for components such as capability, reliability, maintainability, supportability, and producibility, and it is the interplay among them that drives the different levels for both IRL and TRL of the elements in a system. Thus, the optimal selection of which components to enhance to improve the system SRL becomes an optimal system design development problem.

The optimal design of systems is a classical optimization problem in the area of systems engineering. In general, the objective of these problems is to optimize a function-of-merit of the system design (reliability, cost, mean time to failure, supportability, etc.) subject to known constraints on resources (cost, weight, volume, etc.) and/or system performance requirements (reliability, availability, mean time to failure, etc.). To optimize this specific function, it is generally assumed that the system can be decomposed into a system that contains a known number of subsystems or elements (as in Figure 1) and, for each of these elements, a known set of functionally equivalent components types (with different performance specifications) can be used in the design.

From a system engineering design perspective, an optimization approach that balances needs (i.e., the enhancement of the SRL) with resources (i.e., cost of technologies, cost of technology development, etc.), can be an effective and efficient method for reducing risk. That is, the development of a SRL index correlated with the defense acquisition process can be used as an optimization framework for the systems engineer or program manager to design-in enhanced system reliability, maintainability, and supportability to achieve the desired reductions in the necessary logistics footprint and the associated lifecycle cost.

Optimization becomes crucial when trying to decide between competing system design alternatives or when trying to decide which individual TRL or IRL to improve. To make the best decision, optimization models can be developed to assist management to choose SRL improvement opportunities. It is reasonable to assume that improvements will result in costs associated with the purchase of new technology, rework of existing equipment, training of employees, hiring new employees, and enhancements to information technology infrastructure. Two models can be developed. The first model considers minimizing the development cost associated to increasing SRL to some predefined user level, λ. The second model is to maximize the SRL (a function of TRL and IRL) under constraints associated with resources. The mathematical forms of these models follow.

4.1.1 System Cost of Development (SCOD) Minimization

Model SCOD_{min} illustrates an optimization model whose objective is to minimize development cost (a function of TRL and IRL development) under constraints associated with schedule and the required SRL value. The general mathematical form of Model SCOD_{min} follows:

Minimize: SCOD(**TRL,IRL**) = SCODfixed + SCODvariable (**TRL,IRL**)

Subject to: SRL(**TRL,IRL**) ≥ λ

R1 (**TRL,IRL**) ≤ r1

.

.

.

Rh (**TRL,IRL**) ≤ rh

The matrices **IRL** and **TRL** in Model SCOD_{min} contain the decision variables. Each of these variables are integer valued and bounded by (IRL_i,9) and (TRL_i,9), respectively. That is, the TRL/IRL for the i^{th} component cannot be below its current level or above perfect technology development/integration (IRL or TRL = 9).

To completely characterize the decision variables in Model SCOD_{min}, it is necessary to introduce the following transformation:

$$y_i^k = \begin{cases} 1 & \text{If } TRL_i = k \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad x_{ij}^k = \begin{cases} 1 & \text{If } IRL_{ij} = k \\ 0 & \text{otherwise} \end{cases} \quad \text{for } k=1,\ldots 9$$

Notice that based on these binary variables, each of the possible TRL and IRL in the system can be obtained as $TRL_i = \sum_{k=1}^{9} ky_i^k$ and $IRL_{ij} = \sum_{k=1}^{9} kx_{ij}^k$. Based on these binary variables, SRL$_i$ is transformed to

$$SRL_i = \left(\sum_{k=1}^{9} kx_{i1}^k\right)\left(\sum_{k=1}^{9} ky_1^k\right) + \left(\sum_{k=1}^{9} kx_{i2}^k\right)\left(\sum_{k=1}^{9} ky_2^k\right) + \ldots + \left(\sum_{k=1}^{9} kx_{ij}^k\right)\left(\sum_{k=1}^{9} ky_j^k\right) + \ldots + \left(\sum_{k=1}^{9} kx_{in}^k\right)\left(\sum_{k=1}^{9} ky_n^k\right)$$

$$\sum_{j=1}^{n}\left(\sum_{k=1}^{9} kx_{ij}^k\right)\left(\sum_{k=1}^{9} ky_j^k\right)$$

Thus, based on the computation of the SRL with these decision variables, Model SCOD$_{\min}$ belongs to the class of binary, integer-valued, non-linear problems. For a system with n technologies containing m ($m \leq (n-1)n/2$) distinct integrations, and assuming all technologies and integrations are at their lowest levels, there are 9^{n+m} potential solutions to Model SCOD$_{\min}$. Evaluating each possible solution is prohibitive, so to generate an optimal solution faster, a meta-heuristic approach developed by Ramirez-Marquez and Rocco (Ramirez-Marquez & Rocco, 2008) will be applied to the system under development. This approach, called Probabilistic Solution Discovery Algorithm (PSDA), has the capability of producing quasi-optimal solutions in a relatively short period of time. However, it must be mentioned that the results cannot be proven to be the optimal solution. Nevertheless, prior tests have indicated that PSDA results tend to be better than results from alternative meta-heuristic approaches.

4.1.2 SRL Maximization

Model SRL$_{\max}$ follows the same general formulation. It illustrates the optimization model with the objective to maximize the SRL (a function of TRL and IRL) under constraints associated with resources. This model recognizes that the technologies compete for resources and that benefits can result in an improved SRL via the optimal allocation of such resources. The general mathematical form of Model SRL$_{\max}$ is

Model SRL$_{\max}$

$$\begin{aligned}
&\text{Max} \quad SRL\left(\mathbf{TRL}, \mathbf{IRL}\right) \\
&s.t. \\
&R_1\left(\mathbf{TRL}, \mathbf{IRL}\right) \leq r_1 \\
&\vdots \\
&R_H\left(\mathbf{TRL}, \mathbf{IRL}\right) \leq r_H
\end{aligned}$$

The success of implementing these models depends on the consistent and continuous definition of needed capabilities, the maturation of technologies that lead to disciplined development, and the production of systems that provide increasing capability towards a material concept. A fundamental challenge to defense acquisition is that the ultimate functionality cannot be defined at the beginning of the program. Only by the maturation of the technologies, matched with the evolving needs of the user, can they provide the user with capability.

4.2 System Earned Readiness Management (SERM)

The optimization models above can provide valuable insight into the development of a methodology for monitoring and evaluating the overall progress of the development effort. This is primarily due to the fact that the models can identify the optimal development path that can be followed. That is, they identify to what TRL the critical technology elements (CTEs) and which IRL the integration elements should be matured, as well as when those TRLs and IRLs can be achieved.

With these data, we can develop an analytical tool and methodology for evaluating overall progress in systems development as well as measure the impact of alternative or competing architectures, critical technologies and integration elements on the maturity of systems within the systems engineering lifecycle. Furthermore, it can serve as a guide to anticipate the lifecycle implications of the decisions made during the development process. The proposed methodology is termed System Earned Readiness Management (SERM). It will be analogous to Earned Value Management (EVM), an analytical tool used in Project Management (pp. 17-18).

While the optimization models are unavoidably mathematically involved, SERM itself is envisioned to be a relatively simple management tool. It will measure in aggregate terms the level of accomplishment of the system development process. When compared to the development plans and factor estimates that have been prescribed for a particular system under development, management can make conclusions on its status and suggest necessary adjustments to correct any significant deviations. SERM is expected to be valid throughout a wide range of systems with varying degrees of complexity and is intended to be a tool that is easy to use, notwithstanding the complex mathematical algorithms behind it.

Logically, SERM can only be useful if the system under development is already covered by a sufficiently detailed development plan. That is, the system requirements, design and development schedules have already been frozen. However, there are many systems under development that are inherently fraught with high degrees of uncertainty that emanate from the high levels of novelty as well as technology of the system. To be properly managed, such systems have to go through several requirements and design cycles before both can be frozen (Shenhar & Dvir, 2007). However, the need for monitoring and evaluating these systems before the final development cycle still exists. Developing a modified SERM (to be called SERM-U) for such situations will be the ultimate objective.

5. Conclusions

This paper proposes the inclusion of a separate maturity index to measure the progress of the development of the integration links of a system under development. This metric called Integration Readiness Level (IRL) is necessary because in some projects, integration elements have been overlooked and have resulted into major debacles. The paper also introduces the development of a system-focused approach for managing system development and making effective and efficient decisions during the defense acquisition process. For this to be accomplished, a new System Readiness Level (SRL) index will incorporate both the current TRL scale and the proposed IRL metric. The foundations of the SRL are described and we show the techniques for determining current and future readiness of a system to determine its position in the defense acquisition process. In addition, it proposes optimization models than can provide management with an optimal development plan that can meet the objectives of the development team based on constrained resources.

These, in turn, can become the foundation for the development of a monitoring and evaluation tool that will be analogous to Earned Value Management, which is used in project management.

The conceptual development of these metrics and tools outpace their validation and verification in the field. Currently, what is necessary is to have greater involvement from practitioners so that the acquisition community can agree to a common measurement and language that can only improve the system development and acquisition process.

List of References

Barr, Z. (1996, December). Earned value analysis: A case study. *Project Management Network*.

Brandon, D.M., Jr. (1998, June). Implementing earned value easily and effectively. *Project Management Journal*.

Cundiff, D. (2002). *Manufacturing readiness levels (MRL).* DoDs ATD/STO Environments. Unpublished white paper. Carnegie Mellon.

DoD. (2005). *Technology readiness assessment (TRA) deskbook*. Washington, DC: DoD—Science and Technology.

Dowling, T., & Pardoe, T. (2005). *TIMPA technology insertion metrics.* (Vol. 1). Ministry of Defence (Ed.). London: QinetiQ.

GAO. (1999, July). *Better management of technology development can improve weapon system outcomes* (GAO/NSIAD-99-162). Washington, DC: Author.

Gove, R., Sauser, B., & Ramirez-Marquez, J. (2008). Integration maturity metrics: Development of an integration readiness level. *International Journal of Technology Management*. Under review.

Gove, R. (2007). *Development of an integration ontology for systems operational effectiveness* (Master's Thesis). Hoboken, NJ: Stevens Institute of Technology.

Mankins, J.C. (2002). Approaches to strategic research and technology (R&T) analysis and road mapping. *Acta Astronautica, 51*, 3-21.

Meystel, A., Albus, J., Messina, E., & Leedom, D. (2003). *Performance measures for intelligent systems: Measures of technology readiness.* PERMIS '03 White Paper.

Ramirez-Marquez, J.E., & Rocco, C. (2008). All-terminal network reliability optimization via probabilistic solution discovery. *Reliability Engineering & System Safety*. In press.

Sadin, S.R., Povinelli, F.P., & Rosen, R. (1989). The NASA technology push towards future space mission systems. *Acta Astronautica, 20*, 73-77.

Sauser, B.J., Ramirez-Marquez, J.E., Henry, D., & DiMarzio, D. (2008). A system maturity index for the systems engineering life cycle. *International Journal of Industrial and Systems Engineering, 3(6).* In press.

Shenhar, A.J., & Dvir, D.. (2007). *Reinventing project management*. Boston: Harvard Business School Press.

Smith, J.D. (2005). *An Alternative to technology readiness levels for non-developmental item (NDI) software.* Remarks delivered at the 38th Hawaii International Conference on System Sciences, Hawaii.

USD(AT&L). (2005). *The defense acquisition system* (DoD Directive 5000.1). Washington, DC: Author.

Valerdi R., & Kohl. R.J. (2004). *An approach to technology risk management.* Remarks delivered at the Engineering Systems Division Symposium, Cambridge, MA.

Verma, D., Beck, J., & Parry, T. (2003). *Designing and assessing supportability in DoD weapon systems: A guide to increased reliability and reduced logistics footprint*. Washington, DC: OSD, DoD.

Verma, D., Farr, J., & Johannesen, L.H. (2003). System training metrics and measures: A key operational effectiveness imperative. *Journal of Systems Engineering, 6*, 238-248.

Acknowledgements

The authors would like to thank the support of the Naval Postgraduate School Contract # N00244-08-0005, the US Army Armament Research Development and Engineering Center, and Northrop Grumman Integrated Systems for their support in this research.

Panel 6 - Developments in Support of Open Architecture & Service-oriented Architecture Initiatives

Wednesday, May 14, 2008	Panel 6 - Developments in Support of Open Architecture & Service-oriented Architecture Initiatives
1:45 p.m. – 3:15 p.m.	**Chair:** **Mr. William P. Bray**, Program Manager for Integrated Combat Systems, Office of the Program Executive Officer for Integrated Warfare Systems, US Navy **Discussant:** **Mr. Nick Guertin**, Director, Open Architecture, Office of the Program Executive Officer for Integrated Warfare Systems **Papers:** ***AEGIS Platforms: The Potential Impact of Open Architecture in Sustaining Engineering*** **Dr. Thomas J. Housel**, Professor, Naval Postgraduate School ***Which Unchanged Components to Retest after a Technology Upgrade*** **Dr. Valdis A. Berzins**, Professor, Naval Postgraduate School

Discussant: Nick Guertin's duties as the deputy director for OA center on enabling the Navy to buy and build systems as a coordinated enterprise effort. Over the past year, the Naval Open Architecture (NOA) initiative has garnered the attention of both the Chief of Naval Operations (CNO) as well as members of Congress.

Guertin's past duties included chief engineer for submarine combat control, which incorporated the business and technical processes of the use of commercial off-the-shelf (COTS) equipment in the Acoustic Rapid COTS Insertion (ARCI) program. He also served as a systems engineer for submarine sonar (including Acoustic Rapid COTS Insertion sonar system), a heavyweight torpedo depot engineer, and a naval shipyard nuclear test engineer. He is also a retired Navy Reserve officer with submarine service and various engineering duty ship repair and construction assignments leading up to command of a ship repair team.

Which Unchanged Components to Retest after a Technology Upgrade

Presenter: Valdis Berzins is a Professor of Computer Science at the Naval Postgraduate School. His research interests include software engineering, software architecture, computer-aided design, and theoretical foundations of software maintenance. His work includes papers on software testing, software merging, specification languages, and engineering databases. He received BS, MS, EE, and PhD degrees from MIT and has been on the faculty at the University of Texas and the University of Minnesota. He has developed several specification languages, software tools for computer-aided software design, and fundamental theory of software merging.

Valdis Berzins
Computer Science Department
Naval Postgraduate School
Monterey, CA 93943
E-mail: berzins@nps.edu

Abstract

The Navy's open architecture framework is intended to promote reuse and reduce costs. This paper focuses on exploiting open architecture principles to reduce testing effort and costs in cases in which the requirements and code for a subsystem have not been changed, but the code is running on new hardware and/or new operating systems due to a technology-advancement upgrade. This situation is common in Navy and DoD contexts such as submarine, aircraft carrier, and airframe systems, and accounts for a substantial fraction of the testing effort. Unmodified software components need to be retested after a technology upgrade in some, but not necessarily in all cases. This paper reports some early research on conditions under which testing of unmodified components can be avoided after a technology upgrade, outlines an approach for identifying situations in which retesting can be safely reduced, and indicates how to focus retesting in cases in which it cannot be avoided.

Keywords: open architecture, reducing regression testing, automated testing, statistical testing, dependency analysis, reuse, operating system upgrades, hardware upgrades.

1. Introduction

The Navy is implementing the open architecture framework for developing joint interoperable systems that adapt and exploit open system design principles and architectures. Research being performed at the Naval Postgraduate School is pursuing a complementary effort to identify weaknesses and gaps in the current state of knowledge with respect to the development and testing of DoD/DoN systems according to such open systems principles, and to develop or adapt new methods for overcoming those weaknesses. The purpose of this effort is to provide sound engineering approaches to better realize the potential benefits of Navy open architectures and to provide concrete means that support economical acquisition and effective sustainment of such systems.

This project focuses primarily on improving test and evaluation of systems with open architectures, since this aspect can greatly benefit from improvements. Specific goals of this research are to enable the following: (i) reduction of unnecessary testing on every system change, (ii) identification of what specific testing and checking procedures need to be repeated after changes, (iii) limiting the scope of retesting when the latter is necessary, and (iv) enabling a single analysis to provide assurance that all possible configurations that can be generated in a model-driven architecture will satisfy given dependability requirements. This paper reports some preliminary results of this project that address the first three of the goals listed above. A roadmap and technical approach for reaching the fourth goal are outlined in Berzins, Rodriquez and Wessman (2007).

The roadmap provides a long-term plan for eventually eliminating the need for regression testing after each reconfiguration and eventually enabling a “plug-and-fight” capability. This plan depends on the design and certification of a common architecture for a family of systems (FOS) that span a parameterized range of expected requirements, based on detailed standards for the components and connections. In this approach, the architecture is certified to meet its requirements, components are tested against standards and requirement parameters, and reconfiguration is achieved by swapping plug-compatible components with different requirement parameters (2007).

This paper focuses on the shorter-term problem of safely reducing testing for software components whose code has not been changed, without waiting for the results of long-term research and without relying on architecture-level certification.

The motivating context for the work reported here was to increase the effectiveness of quality assurance for Navy technology upgrades. The first step was to investigate conditions under which it is safe to reduce testing for software components whose code has not been changed, so that a larger fraction of the available time and effort could be focused on testing the new functionality introduced by the upgrade.

This focus was adopted after the author interviewed representatives from four of the organizations actually involved in developing such technology upgrades. These interviews indicated (with unanimous support) that those organizations’ highest current priorities are reducing testing for unmodified software components after a technology upgrade and adapting automated testing methods into production use. The initial research, therefore, explored practical methods for checking conditions under which it is safe to reduce or eliminate retesting for unchanged components, and sought solutions that leverage automated testing in the contexts in which it is easiest and most effective to do so.

Technology upgrades are typically performed on a two-year cycle. They often involve migration to the best hardware and operating system version available at the time, where “best” implies a balanced tradeoff between high performance and reliable operation. Typically, only a small fraction of the application code has been changed. However, current certification practices require all of the code to be retested prior to deployment, whether it has been modified or not. Retesting of an unchanged module can be avoided only if we can establish that it has not been adversely impacted by the change. The rest of this paper explores ways to determine that, and the conditions under which such determination is possible.

The rest of this paper is organized as follows. Section 2 describes methods for deciding when re-testing of unchanged components can be safely reduced or eliminated entirely. Section 3 discusses the costs of the automated testing of operating system services needed to support some of the methods presented in Section 2. Section 4 explains the significance of *operational profiles* (probability distributions characterizing expected workloads for software services), which are also needed to support the types of automated testing needed by methods proposed in Section 2. Section 5 identifies the conditions under which unchanged code does need to be tested, along with the potential failure modes that may need to be guarded against and how to focus the retesting to guard against these modes without repeating previous testing effort. Section 6 identifies some relevant previous work, and Section 7 concludes with a summary of the steps that should be taken to enable practical application of the test-reduction approach presented in this paper.

2. Deciding When Retesting Can Be Avoided

If the requirements related to a component have not changed, and the behavior of the components has not changed, then retesting may not be necessary. As discussed further in Section 4 of this paper, the range of conditions under which a component is expected to provide its operational capabilities is a part of its requirements that is particularly relevant to testing and re-testing. The rest of this section addresses how to statically and dynamically check that the behavior of a component has not changed, assuming for the moment that its requirements and range of operating conditions have not changed.

A type of dependency analysis known as program slicing can be used to identify parts of the unchanged code that have the same behavior in the new release as in previous one (Weiser, 1984, July). A program slice at a given observation point is a self-contained subset of the code in the sense that it contains all of the code that can affect the behavior visible at the observation point. If two different programs have the same slice for a given observation point, then they have the same visible behavior at that point. Consequently, if the new release has the same slice as the old release for a given service, then that service will have exactly the same behavior in the new release as in the old one and, consequently, may not need regression testing (Gallagher, 1991, August). This fact is useful because program slices can be computed for software systems on practical (large) scales. The testing-reduction method that follows from this observation is to compute the slice of each service with respect to the new release and the old release, and retest only the services for which these slices differ.

In the context of technology-advancement upgrades, the test-reduction method described above must be augmented with focused, automated testing to produce a substantial reduction in retesting. Technology upgrades usually run on a new version of the operating system. If the source code of the operating system is proprietary and, hence, not available for static analysis (commonly true, except for open source systems such as LINUX), then the only safe assumption is that all operating system services have been impacted by the upgrade to the new version. Thus, any service whose slice includes a dependency on a system call would be potentially impacted and would have to be retested, based on the simple slicing approach outlined in the previous paragraph. This is likely to include most of the application-level modules, thus severely limiting the amount of savings that can be obtained using slicing alone.

Automated testing, however, can enable larger reductions in retesting if it is focused on the middleware interface to the underlying operating system services. Fortunately, the author's interviews with representative stakeholders confirmed that most Navy systems with open architectures are designed around a middleware interface that encapsulates all operating system calls. Such middleware interfaces are also prevalent in other DoD systems, including the US Army's FCS. Application architectures are typically designed in this way to ease the job of porting the application to new operating systems, whether they are new releases of the same product or different products. Consequently, each new release of the operating system and the neighboring middleware layer are both designed to preserve the observable behavior of the previously available system calls if at all possible—even if the details of the implementation may vary from one release to the next. If we know that the observable behavior of a given system call is the same in the old and the new version of the operating system, then we can truncate the slice at the middleware layer for that call, and conclude that the behavior of an application service is unaffected by the OS change if its abbreviated slices in the two versions are the same. The proposed enhancement to dependency analysis using program slicing is to check this property for each system call in the middleware layer via automated testing.

This same strategy can also be applied at higher levels of middleware. For example, for the common case of applications that have been developed for the Java or .NET platforms, the interface to operating system resources is the framework runtime, such as the interface to the Java foundation classes. One related viable strategy for reducing testing of unchanged application code is bounding slicing by the interfaces at this level and using automated testing to show equivalent behaviors of the two releases at these interfaces. A related, common pattern of changes that should not affect behavior involves framework evolution, in which applications are recoded to migrate from “deprecated” (soon to become obsolete) interfaces to the corresponding new versions of the interfaces. Although such changes produce differences in the code, they are intended to preserve behavior, and should be amenable to the automated test strategy. Thus, modules one level above the framework runtime interfaces are additional candidates for automated testing and slicing cutoff boundaries.

Automated testing is attractive in these contexts because a simple, reliable implementation of a “test oracle” is possible for the encapsulated operating systems services. A “test oracle” is a process for automatically determining which test outputs pass and which ones fail. The “unchanged behavior” condition can be easily checked by software for a given set of input data. This is possible since both the old and the new versions of the operating system are available for testing, and test scaffolding software can compare the results of the two versions via equality tests. The existence of such a “test oracle” implies that the OS middleware testing process can be completely automated—enabling economic and practical testing with statistically significant sample sizes that support very high confidence levels, or, in some cases, even exhaustive testing of the operating system interfaces that supports definite conclusions. The proposed automated testing process would, thus, classify all of the services in the middleware interface to the operating system into two groups: those whose behavior is the same in both versions of the operating system (the preserved services), and those whose behavior differs in the two versions (the modified services). We expect the first group to be much larger than the second group.

In such cases, we can cut off slices at the system calls to the preserved services, and conclude that unmodified application components do not have to be retested unless their slices differ or contain system calls that invoke one of the modified services. The operating system interface always needs to be thoroughly retested, but this can be done by the affordable automated process described above.

The above analysis depends on the assumption that we can accept a statistical inference about the unchanged behavior of the operating system’s calls, if the statistical confidence level is high enough. Since most military decisions must be based on information that has the same degree of uncertainty, we do not expect lack of certainty to be a problem in principle. We, therefore, consider how to determine what level of confidence would be "high enough" and how many test cases are necessary to reach that level of confidence.

We start with a consideration that should be meaningful to the stakeholders: if the mean time between observations of a behavioral difference in a given operating systems service is substantially (k times) longer than a mission, it is acceptable to ignore risks due to the possibility of such an unexpected difference. The meaning of "substantially" can be expressed as a numerical safety factor k that can be understood and set by system stakeholders based on their tolerance for risk.

Next, we measure the mean number of executions per mission es for each service s in the middleware interface to the operating system. The objective of the automated testing for each service s is to ensure the mean number of executions between observed differences in the behavior of service s is at least Ns, where

$$\mathbf{N_S = k\ e_S.}$$

Theorem 4.3 from Howden (1987) can then be used to determine the required number of test cases Ts for each service:

$$\mathbf{T_S = N_S \log_2 N_S.}$$

If we run Ts test cases that are independently drawn from the probability distribution characterizing the mission (called the operational profile), the theorem will enable us to conclude that the mean number of executions is at least Ns with a statistical confidence level (1 – 1/Ns); however, this is contingent upon none of the Ts test cases showing any differences in the behavior of the services under the new version of the operating system from those in the previously released version.

The rationale for this choice of confidence level is that it makes the probability of making a false positive conclusion no more than the acceptable frequency of behavioral differences, thus scaling the risk due to random sampling errors to match the specified maximum acceptable failure rate. False positive conclusions correspond to cases in which the frequency of behavioral differences in the new release of the operating system service in question is actually greater than the target bound (1/Ns), but the automated testing procedure failed to observe a difference due to random sampling fluctuations that caused conforming results to appear purely by chance. The test set size Ts has been chosen to make the probability of such a chance observation at most (1/Ns).

Thorough statistical testing of the operating system interfaces has the additional benefits of increasing confidence that differences in hardware (and possibly different versions of the compilers, linkers and loaders) have not affected the behavior of the applications built using these services.

3. Cost of Automated Testing

There are several different kinds of automated testing. The most common kind is semi-automated testing. This approach automates the type of testing currently performed manually. It is commonly the first kind of automated testing implemented in an organization because it does not involve any process changes. In this type of approach, the test cases are still developed individually by test engineers, but the test cases are run automatically, and the results are classified into pass or fail categories automatically—often by comparison to previously captured test outputs that were originally individually examined and categorized by people. In this approach, execution and categorization of test results is automated, but the choice of test cases and the initial pass/fail decisions are not. This approach saves appreciable time and effort relative to a completely manual approach, but the human effort required is still proportional to the number of test cases.

Another approach particularly relevant in our context is automated statistical testing. In this approach, the choice of test cases and the initial pass/fail decisions are automated, as well. This makes a great difference because the human effort involved does not increase with the number of test cases to be executed. This enables economical application of the very large test sets needed to achieve the coverage required to support high levels of statistical confidence in the dependability of the software. The high levels of statistical confidence are needed to avoid testing for other unchanged code based on indirect evidence that the behavior of the underlying services on which the unchanged code depends has not changed.

The context identified in the previous section is well suited for automated statistical testing, because the choice of test cases and the initial pass/fail decisions are easily automated in that context: the first can be done by random sampling from the operational profile, and the second by comparison of the results produced by the previous release of the software to those produced by the new release.

The variation in the number of the test cases Ts required as a function of the acceptable risk of false positive conclusions (1/Ns) is illustrated in Table 1.

Table 1. Number of Test Cases Required for Different Levels of Risk Tolerance

N_s	C	T_s
10^3	.999	1.0×10^4
10^4	.9999	1.3×10^5
10^5	.99999	1.7×10^6
10^6	.999999	2.0×10^7
10^7	.9999999	2.3×10^8
10^8	.99999999	2.7×10^9
10^9	.999999999	3.0×10^{10}

Ns: Desired lower bound on mean number of executions between differences
C: Statistical confidence level
Ts: Number of independent random test cases required

Figure 1 shows how the cost characteristics of the proposed automated testing approach compare to the costs of manual testing. The cost curves are close to straight lines; the fixed costs of automated testing are larger than for manual testing, and the marginal cost of adding another test case is much smaller for automated testing than for manual testing.

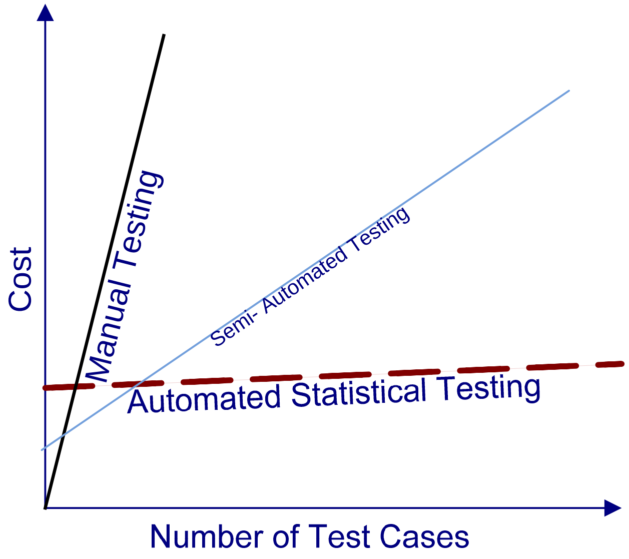


Figure 1. Testing Cost Characteristics

In order to determine the crossover points, we must have experimental data. However, we expect automated testing to be affordable—even for the very large numbers of test cases needed for high confidence in stability of OS services across different releases. We also expect manual and semi-automated approaches not to be affordable when we test to high confidence.

Regarding the time and other resources to perform the proposed automated statistical testing, we can note the following:

1. It typically takes a small amount of time to perform a single system call.
2. Testing using independent, random samples is easily parallelizable and could be effectively spread over large numbers of processors using well-established techniques—such as Google's Map Reduce programming model (Lammel, 2008, January)—if very high confidence levels are needed.
3. Behavior of operating system calls can be tested independently of other shipboard systems and does not require interactions with human operators.

Since the testing process is completely automated, the variable cost of these tests is due to computing time and hardware, but not to human effort. The benefit of the automated statistical test approach described here is that there are no variable costs for labor. Since computing resources are currently inexpensive and steadily getting cheaper, this implies that even the relatively large numbers of test cases needed for high confidence are likely to be affordable.

This approach does involve some fixed costs for human effort that may be higher than in less-disciplined manual approaches. These costs are due to the need for the following activities:

1. Measurement of operational profiles—i.e., the frequency distributions of operating system calls and their associated input parameters. Instrumented versions of the

software can be used during exercises to collect measurements of the operational profiles, or, if the computational overhead of doing this is acceptable, measurements could also be collected during actual operations.

2. Coding more sophisticated test-driver software that includes code for generating random samples from the measured operational profiles, code that implements test oracles as described in Section 2, and code that keeps track of testing statistics and reports them.

4. Why Do We Care about Operational Profiles?

Accurate estimates of operational profiles, preferably based on actual measurements, are necessary because in all practical cases, the reliability of a software system is meaningless without firm knowledge of the operational profile. This claim is based on the hypothesis that all real systems have at least one input value x for which they perform correctly, and at least one other input value y for which they do not. If we know x and y, we can construct a spectrum of possible operational profiles for which the reliability of the same system ranges from 0 to 1 and attains every value in between.

The above line of reasoning shows that the only systems whose reliabilities do not depend on the operational environment are those that fail for all possible inputs (reliability uniformly 0, not interesting), and those that operate correctly for all possible inputs (reliability uniformly 1, not attainable in practice for large systems).

For all other systems, the reliability is determined by the operational profile and can vary widely for different operational contexts. This has serious implications for component reuse, which is a cornerstone of the open Navy architecture initiative.

Operational profiles have been used by the testing research community for many years and have been applied in many contexts. For example, they have been measured and used to assess the reliability of telephone-switching software.

5. When Retesting Is Needed

If the process described in Section 2 shows that the slice of a given application level service differs in the new and previous release, then behavior of the system has been impacted, and the service needs retesting. Services whose requirements have changed will be in this category—so new functionality needs to be tested according to the criteria proposed in Section 2, as expected. If the behavior of unchanged modules with unchanged requirements can be affected by other modified modules, they will be also identified by the slicing process. These also need retesting to check if there are any unintended indirect consequences of the code changes. This is the effect most developers and test and evaluation organizations are concerned about guarding against.

In addition, however, some modules may need to be retested even if their behavior has not changed, because reliability of a system depends on its environment as well as on its implementation. Thus, changes to the environment of a system can affect its reliability even if the behavior of the system remains unchanged. This possibility must be considered in the context of Navy open architectures because they strongly encourage reuse of components in a variety of operational environments to provide cost savings.

When a reusable component is moved unchanged to a new operating environment, we need to check whether its range of expected operating conditions has expanded—as manifested by an expanded range of expected input parameter values in its new operating environment in comparison to its previous operating environment. If this is the case, then the component needs to be tested both on samples from the previously untested part of the input space, as well as on scenarios typical of the novel features of the new operating environment for the reusable component. If the analysis of the operating environment is done properly, we will not have to repeat the tests conducted previously, but rather must run new and substantially different tests that focus on the new situations that are likely in the new operating environment but were not likely in the previous ones.

One other issue to be considered is whether the requirements of the component involve timing constraints. The above discussion has focused mostly on the functional behavior of the component, and not on how much time it takes to produce those results. Components that are subject to strict timing requirements need additional quality-assurance; analysts must check those requirements if the characteristics of the hardware in the new release will differ from those in the old one. Perhaps surprisingly, this is the case even if the new hardware is faster than the old hardware. This is due to the properties of the scheduling methods to be used. In particular, it is known that rate-monotonic scheduling, one popular method for scheduling real-time software, is (in some cases) subject to anomalies. For instance, a given schedule may work fine for a given hardware configuration but may miss deadlines when executed on faster hardware. This can happen if uninterruptible operations or those that lock shared resources are executed in a different order on the new hardware—due to the completion of a sped-up task prior to the release time of a competing task that was previously unreachable. Methods for checking dependencies on timing constraints are beyond the scope of this paper.

To focus retesting where it is needed most, the author recommends the establishment of an explicit process to track past and projected changes in operational profiles and to reflect these changes in testing plans. Some preliminary steps in this direction are to:

1. Keep records of operational profiles used in testing previous releases of subsystems.
2. Measure operational profiles under mission conditions and exercises exploring new concepts of operations. Check for differences from those covered in past testing.
3. Focus retesting efforts on circumstances and scenarios that have weight in actual and projected operational profiles that have not been covered well in previous testing of the same unchanged components.

6. Relevant Previous Work

Program slicing has been used in a wide variety of applications, including testing (Binkley, 1998; Gupta, Harrold & Soffa, 1992; Harman & Danicic, 1995; Hierons, Harman & Danicic, 1999; Hierons, Harman, Fox, Ouarbya & Daoudi, 2002), debugging (Agrawal, DeMillo & Spafford, 1993; Lyle & Weiser, 1987), program understanding (De Lucia, Fasolino & Munro, 1996; Harman, Hierons, Danicic, Howroyd & Fox, 2001), reverse engineering (Canfora, Cimitile & Munro, 1994), software maintenance (Gallagher, 1991, August; Cimitile, De Lucia & Munro, 1994), change merging (Horwitz, Prins & Reps, 1989; Berzins & Dampier, 1996), and software metrics (Lakhotia, 1993; Bieman & Ott, 1994). More detailed surveys of previous work on slicing can be found in Binkley and Harmon (2004). Although the subject is outside the scope of the current paper, which focuses on reducing testing by detecting unintended interactions between different parts of a program, Gallagher (1991, August) also outlines a method for preventing the introduction of new unintended interactions during software upgrades.

Automated testing has been studied in a wide variety of contexts. An approach to automatically generating test-driver code from formal requirements is described in Berzins and Chaki (2002). This approach automatically generates open sets of test cases based on random samplings from implementations of operational profile distributions. The pass/fail decisions that classify the results produced by the individual test cases are made by software methods that are automatically generated from the requirements, which must be sufficiently precise and constructive to support this process. The number of test samples in the generated test set is automatically set to meet specified reliability goals expressed in terms of mean number of executions between failures. This work provides an approach to extending automated statistical testing to contexts beyond those in which the expected behavior of a module is unchanged in the new release.

There has also been previous work on quality assurance for flexible systems at the levels of requirements (Luqi, Zhang, Berzins & Qiao, 2004, December; Luqi & Lange, 2006, November 8) and architectures (Berzins & Luqi, 2006, May 6; Luqi & Zhang, 2006, May 6). In addition, a method for assessing the impact of timing constraints on reliability of system upgrades can be found in Qiao, Wang, Luqi, and Berzins (2006, March).

7. Conclusion

Further research is recommended to substantiate the practical applicability of the ideas outlined above. Experimental evaluation of the slicing method for identifying modules that do not have to be rested should be performed, together with the focused automated testing methods needed to fully realize the potential savings of the approach.

Measurement and analysis of the operational profiles of reusable components can be used to support analysis of changes in the operating environment that may require focused retesting of components whose behavior has not changed. Operational profiles are probability distributions that serve as mathematical representations of the operating environment and that are needed to support statistically significant testing that can reduce the testing effort, as described above. These distributions can be measured by instrumenting components and collecting statistics as they run, either in exercises or during actual missions, and can be used to drive statistically based automated testing that can quantitatively assess the reliability of systems.

Although it is not easy to convince contractors to automate their testing if they are not familiar with this approach, the economic incentives to do so are getting more compelling. This practical problem is particularly evident in the current situation—in which domain experts are often doing the project management and coding with little knowledge of or experience with recent advances in the techniques and tools used in software engineering. The increasing popularity of agile methods, which depend heavily on semi-automated testing, should help change this perception. Pilot projects demonstrating the effectiveness of the suggested approach are recommended to provide concrete data about costs and benefits, thereby alleviating concerns about project risks due to technology innovations.

List of References

Agrawal, H., DeMillo, R., & Spafford, E. (1993). Debugging with dynamic slicing and backtracking. *Software Practice and Experience, 23*(6), 589–616.

Berzins, V., & Chaki, N. (2002, October 7-11). MTDOAG: Module Test Driver and Output Analyzer Generator. In *Proceedings of the 9th Monterey Workshop: Radical Innovations of Software and Systems Engineering in the Future* (pp. 48-56). Venice: Universita Ca Foscari di Venezia.

Berzins, V., & Dampier, D. (1996). Software merge: Combining changes to decompositions. *Journal of Systems Integration, 6*(1-2, special issue on Computer-aided Prototyping), 135-150.

Berzins, V., & Luqi. (2006, May 6). Achieving dependable flexibility via quantifiable system architectures. In *Proceedings of Workshop on Advances in Computer Science and Engineering* (pp. 53-54). Berkeley, CA .

Berzins, V., Rodriguez, M., & Wessman, M. (2007, May 16-17). Putting teeth into open architectures: Infrastructure for reducing the need or retesting. In *Proceedings of the Fourth Annual Research Symposium—Acquisition Research: Creating Synergy for Informed Change* (pp. 285-311). Monterey, CA: Naval Postgraduate School.

Bieman, J., & Ott, L. (1994). Measuring functional cohesion. *IEEE Transactions on Software Engineering, 20*(8), 644–657.

Binkley, D. (1998). The application of program slicing to regression testing. In M. Harman & K. Gallagher (Eds.), *Program Slicing, Information and Software Technology, 40*(11-12), (pp. 583-594) (special issue).

Binkley, D.W., & Harman, M. (2004). A survey of empirical results on program slicing. In M.V. Zelkowitz (Ed.), *Advances in Computers* (pp. 105–178). (Vol. 62). San Diego, CA: Elsevier.

Canfora, G., Cimitile, A., & Munro, M. (1994). RE^2: Reverse engineering and reuse re-engineering. *Journal of Software Maintenance: Research and Practice, 6*(2), 53–72.

Cimitile, A., De Lucia, A., & Munro, M. (1996). A specification driven slicing process for identifying reusable functions. *Software Maintenance: Research and Practice, 8*(3), 145–178.

De Lucia, A., Fasolino, A., & Munro, M. (1996). Understanding function behaviours through program slicing. In *Proceedings*, *4th IEEE Workshop on Program Comprehension* (pp. 9–18). Los Alamitos, CA: IEEE Computer Society Press.

Gallagher, K. (1991, August). Using program slicing in software maintenance. *IEEE Transactions on Software Engineering, 17*(8), 751-760.

Gupta, R., Harrold, M., & Soffa, M. (1992). An approach to regression testing using slicing. In *Proceedings of the IEEE Conference on Software Maintenance* (pp. 299–308). Los Alamitos, CA: IEEE Computer Society Press.

Harman, M., & Danicic, S. (1995). Using program slicing to simplify testing. *Software Testing, Verification and Reliability, 5*(3), 143–162.

Harman, M., Hierons, R., Danicic, S., Howroyd J., & Fox, C. (2001). Pre/post conditioned slicing. In *Proceedings*, *IEEE International Conference on Software Maintenance ICSM'01* (pp. 138–147). Los Alamitos, CA: IEEE Computer Society Press.

Hierons, R., Harman, M., & Danicic, S. (1999). Using program slicing to assist in the detection of equivalent mutants. *Software Testing, Verification and Reliability, 9*(4), 233–262.

Hierons, R., Harman, M., Fox, C., Ouarbya, L., & Daoudi, M. (2002). Conditioned slicing supports partition testing. *Software Testing, Verification and Reliability, 12*(1), 23–28.

Horwitz, S., Prins, J., & Reps, T. (1989). Integrating non-interfering versions of programs. *ACM Transactions on Programming Languages and Systems, 11*(3), 345–387.

Howden, W. (1987). *Functional program testing and analysis*. New York: McGraw-Hill.

Lakhotia, A. (1993). Rule-based approach to computing module cohesion. In *Proceedings of the 15th Conference on Software Engineering, ICSE-15* (pp. 34–44). Los Alamitos, CA: ACM/IEEE.

Lammel, R. (2008, January). Google's map reduce programming model—Revisited. *Science of Computer Programming*, *70*(1), 1-30.

Lyle, J., & Weiser, M. (1987). Automatic program bug location by program slicing. In *Proceedings*, *2nd International Conference on Computers and Applications* (pp. 877–882). Los Alamitos, CA: IEEE Computer Society Press.

Luqi, & Lange, D. (2006, November 8). Schema changes and historical information in conceptual models in support of adaptive systems. In *Proceedings*, *First International Workshop on Active Conceptual Modeling of Learning* (pp. 112-121)*.* Tucson, AZ: Springer LNCS 4512.

Luqi, & Zhang, L. (2006, May 6). Documentation Driven Evolution of Complex Systems. In *Proceedings of Workshop on Advances in Computer Science and Engineering* (pp. 141-170). Berkeley, CA.

Luqi, Zhang, L., Berzins, V., Qiao, Y. (2004, December). Documentation driven development for complex real-time systems. *IEEE Transaction on Software Engineering, 30*(12), 936-952.

Qiao, Y., Wang, H., Luqi, & Berzins, V. (2006, March). An admission control method for dynamic software reconfiguration in complex embedded systems. *International Journal of Computers and Their Applications, 13*(1), 28-38.

Weiser, M. (1984, July). Program slicing. *IEEE Transactions of Software Engineering, SE-10*(4), 352-357.

THIS PAGE INTENTIONALLY LEFT BLANK

Panel 7 - Considerations for Controlling Lifecycle Costs

Wednesday, May 14, 2008

1:45 p.m. – 3:15 p.m.

Panel 7 - Considerations for Controlling Lifecycle Costs

Chair:

Mr. John Birkler, Research Leader, Acquisition, Defense Production, and Technology Base Issues, RAND Corporation

Discussant:

CAPT Mike Moran, USN, Maritime Patrol & Reconnaissance Aircraft, P-8A Poseidon Program Manager

Papers:

Case Study and Analysis of an Organic and Contractor Logistics Support (CLS) Hybrid Maintenance Model for the Lifecycle of the P-8 Poseidon

LCDR Shane Tallant, USN, LT Mike Martin, USN, and LCDR J. Scott Hedrick, USN, Graduate Students, Naval Postgraduate School

Stryker Suitability Challenges in a Complex Threat Environment

Dr. Don McKeon and **Dr. Paul Alfieri**, Defense Acquisition University

Chair: John Birkler is a senior management system scientist at RAND. He is responsible for managing US Navy, US Coast Guard, and UK Ministry of Defence research projects. He holds a BS and MS in Physics, and completed the UCLA Executive Program in Management in 1992. After completing his third Command tour, he retired from the Navy Reserve with the rank of Captain.

Striker Suitability Challenges in a Complex Threat Environment

Presenter: Dr. Paul Alfieri is a professor of Engineering Management at the Defense Acquisition University (DAU) and is currently serving as the Director of Research. Following 20 years in the US Navy—in which he was a helicopter pilot and engineering duty officer—and several years in the defense industry, he joined the DAU faculty in 1991. Dr. Alfieri has provided engineering consulting support to the Naval Air Systems Command and the Navy Research Laboratory for various projects. He has earned an MS (Aerospace Engineering) from the Naval Postgraduate School, an MSA from George Washington University, and a PhD from Virginia Polytechnic Institute and State University.

Paul Alfieri
E-mail: paul.alfieri@dau.mil

Author: Dr. Donald McKeon is a Defense Acquisition University (DAU) professor of Systems Engineering Management and is located in Warren, MI, supporting the Tank-Automotive and Armaments Command–Life Cycle Management Command (TACOM–LCMC). Dr. McKeon teaches systems engineering, manufacturing and quality, software acquisition management, program management, and performance-based contracting. He has consulted on programs for the Air Force and the Army. Prior to joining DAU, Dr. McKeon worked in research at Sandia National Laboratories and in product development and manufacturing in private industry. Dr. McKeon has a BS, MS, and PhD in engineering and an MBA.

Donald McKeon
E-mail: don.mckeon@dau.mil

Abstract

The cost of operating and maintaining weapon systems is a large expense to the Department of Defense (DoD), and suitability performance is a major factor affecting these costs. Systems with poor suitability performance (such as low reliability, high failure rates, high spare parts usage, and low availability) are extremely difficult to support in a constrained resource environment. For many DoD acquisition programs, suitability lags effectiveness during program development. Suitability determinants (such as reliability and maintainability) are generally not addressed early enough during program development (prior to fielding) and are not prioritized with the same vigor and discipline as performance parameters like speed, accuracy, and lethality. The JROC, DOT&E, and USD(AT&L) have each called for increased attention to suitability improvement.

Introduction

The primary purpose of this article was to investigate the suitability performance challenges of the recently deployed Stryker system, which was accelerated into combat in 2003. Suitability drivers were identified and possible causal factors were investigated. Several specific suitability issues for the Stryker system were revealed during this study. Stryker is performing well in the field with an Operational Readiness Rate (ORR) consistently above the required contractual value. However, a harsh combat scenario, dynamic threat environment, and extremely high tempo of operations have created unique challenges to operators and maintainers.

Background

During his first annual report to Congress, the newly confirmed Director of Operational Test and Evaluation (DOT&E) Dr. Charles E. McQueary made three initial observations. His first observation was that Operational Test & Evaluation (OT&E) is too often the place where performance deficiencies are discovered. Finding performance problems early in the Department of Defense (DoD) acquisition process is important—either in government Developmental Test & Evaluation (DT&E) or contractor testing. Detecting and correcting design issues early in the development process will mitigate program cost overruns and schedule delays. McQueary’s second observation was that the DoD acquisition system is inherently slow and must improve to accommodate rapid fielding of new weapons systems and new technologies. The need for rapid fielding of new technology is evident in the extended hostilities in Iraq and Afghanistan (e.g., armor upgrades for the High Mobility Multipurpose Wheeled Vehicle (HMMWV) and the new Mine Resistant Ambush Protected (MRAP) vehicle). His third observation was that operational suitability of DoD systems is too low and needs to improve. The definition of operational suitability, which can be found in the *Defense Acquisition Guidebook*, Chapter 9 (Operational Test and Evaluation), Section 9.4.5 (Evaluation of Operational Suitability), is as follows:

> Operational Suitability is the degree to which a system can be satisfactorily placed in field use, with consideration given to reliability, availability, compatibility, transportability, interoperability, wartime usage rates, maintainability, safety, human factors, manpower supportability, logistics supportability, documentation, training requirements, and natural environmental effects and impacts. (Duma & Krieg, 2005)

The Cost of Low Suitability

Low suitability is a direct contributor to higher lifecycle support costs. Data for the previous three years (2004–2006) showed that 35% of Initial Operational Test & Evaluation (IOT&E) phases resulted in unfavorable suitability evaluations as reported to Congress in each system’s Beyond Low Rate Initial Production (BLRIP) Report (Director, Operational Test and Evaluation, 2007).

While the technical performance of weapon systems (such as speed, accuracy, and firepower) has improved significantly over the last several decades, suitability parameters (such as reliability, availability, and maintainability) have not improved. Figures 1, 2, and 3 indicate that this problem has been a trend for more than 20 years. All data in Figures 1–3 are based on Army Test and Evaluation Command (ATEC) programs evaluated during the years shown. Figure 1 (Duma & Krieg, 2005) shows that from 1985 through 1990, only 41% of programs evaluated by ATEC successfully demonstrated reliability requirements during operational testing. Figure 2 (Duma & Krieg, 2005) shows that between 1996 and 2000, only 20% of programs met reliability requirements; and Figure 3 (US Army Test and Evaluation Command, 2007) shows that from 1996–2005, only 34% of programs met reliability requirements.

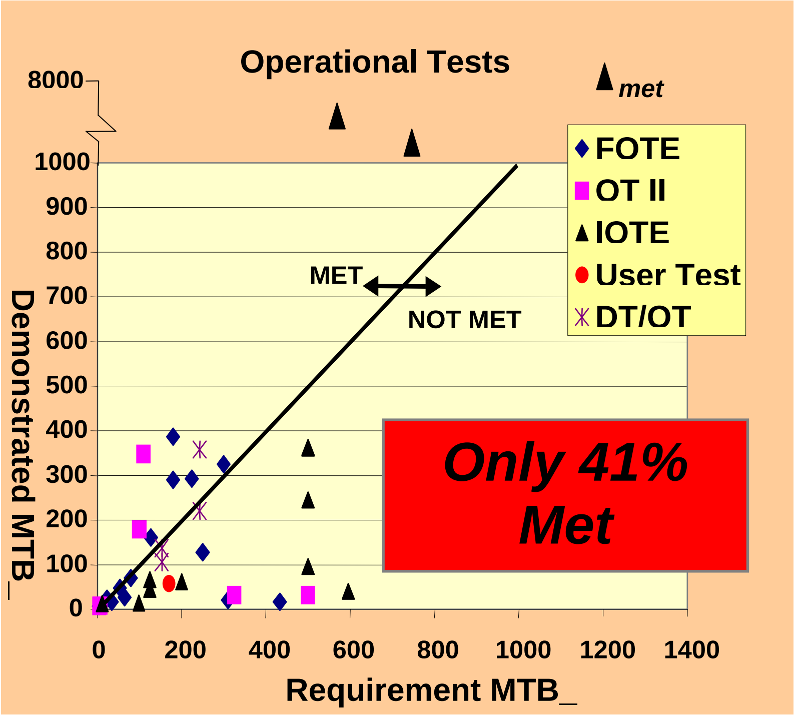


Figure 1. Reliability During Operational Tests (1985–1990)
(Duma & Krieg, 2005)

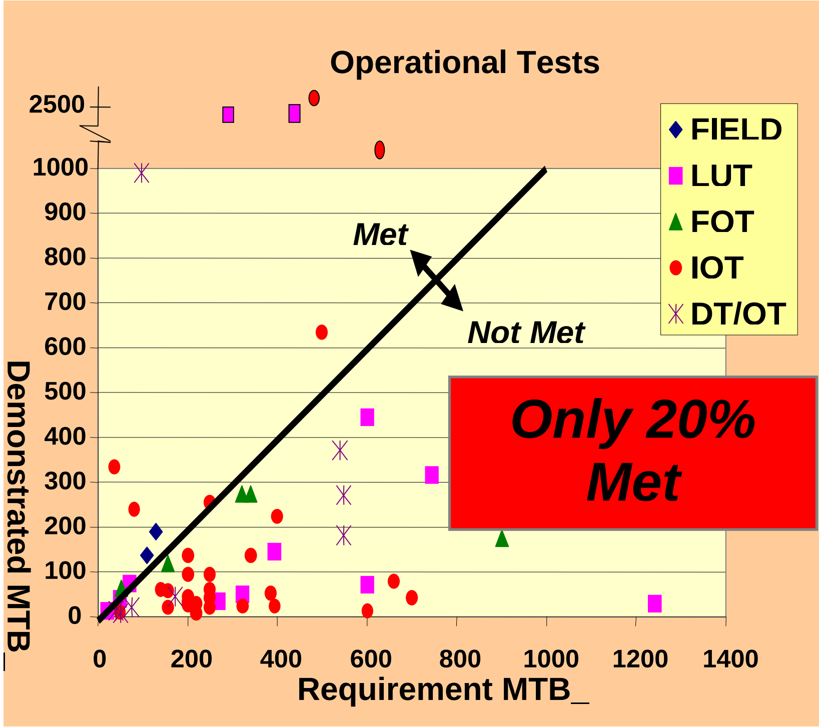
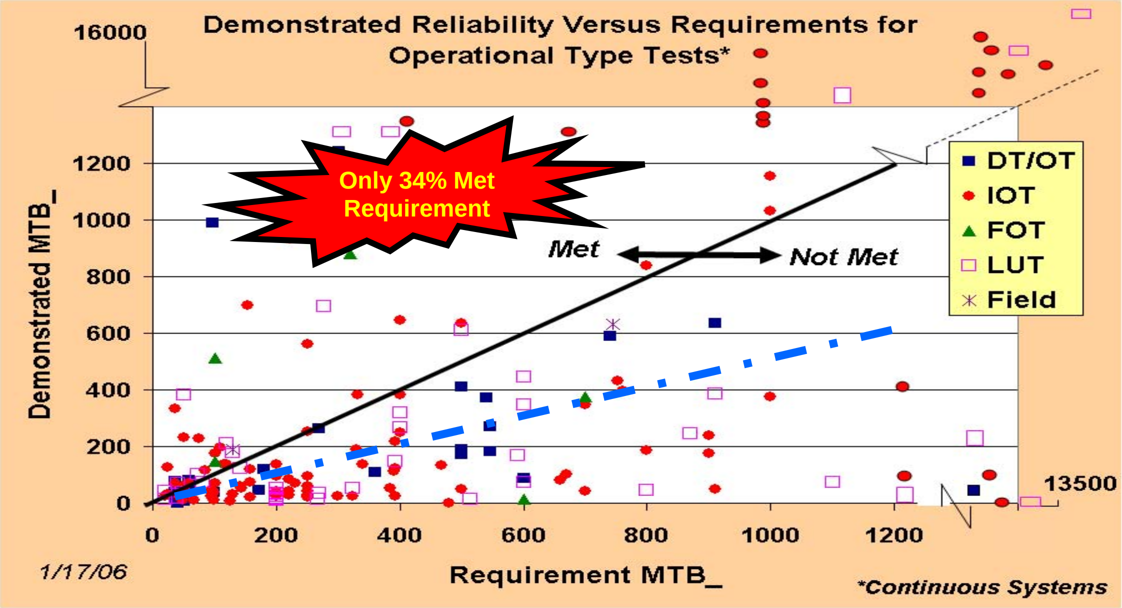


Figure 2. Reliability During Operational Tests (1996–2000)
(Duma & Krieg, 2005)



Amongst Systems Which Did Not Meet Reliability Requirements in OT, 75% of Them Failed to Achieve Half of Their Requirement

Figure 3. Reliability During Operational Tests (1996–2005)
(US Army Test and Evaluation Command, 2007)

Stryker was a new Army program in 2000, but suitability issues were certainly not a new problem. The Defense Science Board (DSB) pointed out in 2000 that 80% of US Army defense systems fail to achieve even half of their required reliability parameters (National Research Council, 2006). Steps have been taken to help address this concern. In November 2004, the Under Secretary of Defense for Acquisition, Technology and Logistics (USD(AT&L)), directed that acquisition programs measure performance in terms of operational availability, mission reliability, and cost per unit of usage (USD(AT&L), 2004). Three months later, the USD(AT&L) issued a memorandum on Total Life Cycle Systems Management (TLCSM) Metrics, which provided specific definitions, formulas, and metrics for calculating important suitability parameters, such as operational availability and mission reliability. In 2005, the DSB recommended that the DoD aggressively pursue implementation of performance-based logistics for all weapon systems. The USD(AT&L) has also directed that the TLCSM Executive Council develop a metrics handbook to be used in performance-based contracts and sustainment oversight (USD(AT&L), 2004; 2006). In August 2006, the Joint Requirements Oversight Council (JROC) mandated a Key Performance Parameter (KPP) of *materiel availability* including key system attributes of *materiel reliability* and *ownership costs* (Joint Requirements Oversight Council, 2006). These initiatives were designed to improve operational performance, establish standard suitability metrics, and reduce lifecycle support costs of new DoD weapon systems.

McQueary’s third observation in his FY 2006 Annual Report is the basis for this research article. Many times, systems receiving favorable effectiveness evaluations but unfavorable suitability evaluations from IOT&E are fielded before suitability shortcomings are corrected. Even though there may be good reasons for deploying these systems before correcting all suitability issues (such as an urgent combat need or the negative consequences of stopping a hot production line), fielding systems before suitability deficiencies are corrected will result in reduced operational availability and increased support costs. Low suitability directly results in increased lifecycle support costs. These costs can appear in many forms, such as increased spares, increased contractor support, increased maintenance actions, increased maintenance man-hours, decreased reliability, decreased availability, and decreased combat capability. Costs over and above the planned costs of lifecycle support can represent a large and unbudgeted expense for the DoD. This undesirable trend of low suitability during major weapon system development has been observed for at least 20 years across all services, and this trend is not improving. For example, the reliability success rate of Army systems tested in 1996–2005 (34%) is lower than the reliability success rate for 1985-1990 (41%).

Overview

The Stryker family of vehicles was conceived as part of the Army's Transformation Campaign Plan. In 1999, General Eric Shinseki, the Army Chief of Staff, came to the conclusion that the Army had serious deployability and mobility issues (Military.com, 2007). Though the Army was capable of full-spectrum dominance, its organization and force structure were not optimized for strategic responsiveness. Army light forces could deploy rapidly, but they lacked the lethality, mobility, and staying power necessary to be effective in peacekeeping scenarios. On the other hand, Army mechanized forces possessed the necessary lethality and staying power but required a large logistics footprint, which hindered their ability to be quickly deployed.

Subsequently, the Secretary of the Army announced a new Army vision in October 1999 to build a landpower force capable of strategic dominance across the full spectrum of ground combat operations. The key to implementing this vision was that the Army become more strategically responsive. Stryker was designed as a full-spectrum, early-entry combat force and optimized primarily for employment in small-scale contingencies. It was developed to operate in a complex environment, including urban terrain, while confronting low- to mid-range threats with conventional and asymmetric capabilities. Requirements for the Stryker include rapid deployment, early entry execution, and the ability to conduct effective combat operations immediately upon arrival (Training and Doctrine Command, 2000, June 30).

Schedule-driven Compromises

Stryker was initially deployed to Iraq in 2003 due to an urgent combat requirement. Prior to deployment, Stryker underwent an aggressive and accelerated development and test program. The urgency of the war prevented the complete spectrum of operational testing to be completed within allowable time constraints. During IOT&E, only a few selected missions, types of terrain, and levels of conflict intensity were evaluated. Also, vehicles used did not accrue sufficient operating time to yield statistically relevant Reliability and Maintainability (R&M) data. In addition, a major configuration change was not included as part of IOT&E or PVT (Production Verification Tests) because add-on armor was not available for installation when testing was performed. The add-on armor package increased vehicle weight by approximately 20%. Since these tests were done in under-stressed conditions (without add-on armor), long-term durability problems were unlikely to be detected (National Research Council, 2004).

Schedule-driven compromises in T&E are not unusual to DoD programs.

> Pressures on program officials to meet budgets and deadlines, due to congressional and other oversight, result in test strategies geared toward demonstrating "successful" performance. Thus, testing is often carried out under benign or typical stresses and operating conditions, rather than striving to determine failure modes and system limitations under more extreme circumstances. (National Research Council, 2006, p. 19)

According to an article printed in the *Detroit News* (Zagaroli, 2005), the Project on Government Oversight, a nonprofit government accountability organization, reported that Stryker was rushed through development, and lack of complete testing could give operators a false sense of security if failure modes are not understood (2005). However, the same newspaper article acknowledged that reports from the field overwhelmingly indicated that Stryker was performing in an outstanding manner. One of the early decisions made by the Army to support an accelerated development and deployment timeline was to rely on contractor performance-based logistics (PBL) support within the Stryker brigades. Some of the duties of the contractor personnel included conducting maintenance on the Stryker vehicle and managing the Stryker-specific supply chain. When Stryker was first deployed to Iraq, the Army did not have the institutional capability to train soldiers on conducting Stryker vehicle maintenance, and therefore faced an immediate need for contractor maintenance personnel to support the deployment (GAO, 2006, September 5).

Each deployed Stryker brigade was fielded with 45 imbedded vehicle maintenance contractor personnel. The Army desires to eventually replace the 45 contractors with active duty soldiers. Current plans call for implementation (removal of embedded contractors) to begin in 2008; however, the GAO reported that this goal will be difficult for the Army to achieve for several reasons. First, the 45 imbedded contractor maintenance personnel must be replaced by 71 soldiers due to other collateral duties and common training requirements of soldiers. Second, the Army is very short of personnel with the five military occupational specialties for wheeled vehicle mechanics—resulting in a very difficult recruiting challenge for the Army. Currently, as reported by the *Washington Post* (White, 2007) and the *New York Times* (Cloud, 2007), the Army is falling short of current recruiting goals.

Operational Readiness

A key factor affecting Stryker suitability performance is deployed operational tempo (OPTEMPO). The program office estimates that the operational tempo is 6 times greater than the originally planned OPTEMPO. Other interviews yielded estimates of operational tempo up to 10 times the planned OPTEMPO. Harry Levins (2007) reports that vehicles in Iraq are using up 7 years of service life for each year of service in Iraq. The Government Accountability Office (GAO, 2006, September 5) estimates that service life is being expended 800% faster than expected. This greatly increased operational tempo results in unexpected failure modes and increased failure rates.

A general finding of this study was that the Army is satisfied with Stryker's performance in the field. System performance in an asymmetric combat scenario under difficult environmental conditions exceeds Army expectations. Brigade commanders have consistently reported high operational readiness rates (greater than 90%) since Stryker was fielded, despite the fact that combat conditions in Iraq have been much different than expected (Figure 4). For example, from October 2003 to September 2005, the first two Stryker brigades that deployed to Iraq reported an average Operational Readiness Rate (ORR) of 96%, which was well above the Army-established ORR performance goal of 90%.

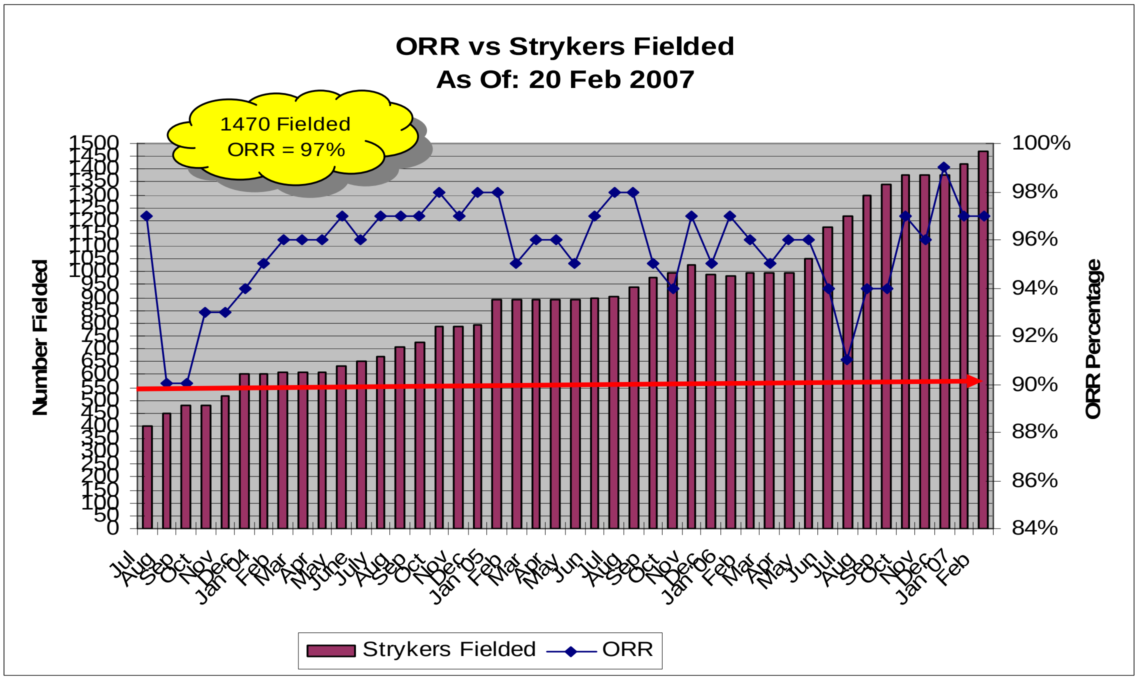


Figure 4. Operational Readiness Rates

Due to the asymmetric nature of the threat forces and to the highly adaptive nature of the enemy, the combat scenarios and operating environment have been much different than expected. According to the Stryker Interim Armored Vehicle Operational Mode Summary/Mission Profile (IAV OMS/MP) (Training and Doctrine Command, 2000), the Stryker planned mission profile called for operations on hard roads 20% of the time, and cross-country operations 80% of the time. The actual Stryker usage in Iraq has been almost the opposite (~ 80% on hard roads, 20% cross-country). Most missions resemble police actions in an urban environment on paved roads, and crews must routinely drive over curbs and other small obstacles to navigate the urban environment. This requires a higher tire pressure than normal, causing more vibration and shock loads and high structural stress on the vehicles.

In response to the greater threat of rocket propelled grenades (RPGs), improvised explosive devices (IEDs), and small projectiles, the Army configured Stryker with an add-on slat armor package and crews added sand bags. The additional weight affected the performance of the Stryker family of vehicles as follows:

- To operate with the increased vehicle weight, the operating tire pressure had to be increased from the design specification of 80 psi to 95 psi. Stryker is configured with a centralized tire pressure system that is designed to automatically keep the tire pressure at the optimum value for specific terrain conditions, speed, and traction. The automatic inflation system was not designed to maintain 95 psi, so soldiers must set tire pressure manually and check it three times daily (Smith, 2005). The requirement to over-inflate the tires to 95 psi and to physically check tire pressure three times per day is an operational nuisance because these are unplanned, but necessary, preventive maintenance actions. Additionally, the combination of routine excessive structural stress and increased tire pressure causes unanticipated structural failures. For example, a large number of wheel spindles developed fatigue cracks and had to be replaced early. Drive shafts are also failing sooner than expected.

- Due to the issues of added weight, excessive tire pressure, and severe operating conditions, tires are also failing at a high rate. In one 96-hour test period at Fort Irwin, CA—with 16 Stryker vehicles—13 tires had to be changed (*WorldNetDaily*, 2003). The *Washington Post* reported that 11 tire and wheel assemblies fail every day, and the GAO asserts that each Stryker vehicle is going through one tire per day on average (Smith, 2005). The additional maintenance actions (checking/adjusting tire pressures and changing tires) are extremely burdensome to the crews since changing tires is not crew-level maintenance and requires special tools.
- The 5,000 pounds of armor to counter RPG threats is generally effective but has many negative operational consequences, such as limited maneuverability, increased component stresses, safety issues, and transportability issues. The extra weight and increased physical dimensions caused by the add-on slat armor adversely impacts performance, especially when maneuvering in spaces with narrow clearance and maneuvering in wet conditions. Operations in soft sand or wet conditions (mud) place additional stress on engines, drive shafts, and differentials; these items have experienced higher-than-normal failure rates (Dougherty, 2004).
- Also, the slat armor causes multiple problems for safe and effective operations. Slat armor can deform during normal operations, sometimes blocking escape hatches and the rear troop egress door. The armor adds approximately 3 feet to the vehicle’s width and can interfere with the driver’s vision. Armor also makes it difficult for others to see the Stryker at night, which is a safety hazard in the urban environment. The armor is very heavy for the rear ramp and strains lifting equipment; crews must sometimes manually assist raising or lowering the rear ramp. The armor attaching bolts on the rear ramp can break off with normal use (increasing the maintenance burden) and may generate unsafe conditions. In addition, slat armor prohibits normal use of storage racks, which may impact operations. Lastly, slat armor affects the transportability of the vehicle in a C-130 cargo aircraft, since the extra weight greatly reduces transport range (GAO, 2004).

Even though these operational issues caused by the add-on slat armor place additional maintenance burdens on crews, Stryker has been reported to be well-suited for the urban fight. Unlike the M-1 tank, Stryker can operate very quietly at high speed, which can be a tremendous tactical advantage (Tyson, 2003). Most Army personnel interviewed felt strongly that Stryker’s tactical performance in the urban environment in Iraq was significantly better than the M113A3, HMMWV, Bradley Fighting Vehicle, or Abrams Tank.

In response to unanticipated urgent combat needs in Iraq, some engineering improvements (configuration changes) were performed on the Stryker since deployment. Since the Army did not buy the technical data package because of its cost, these engineering changes have resulted in increased costs and potential risks (GAO, 2006, July). The GAO reports that current DoD acquisition policies do not specifically address long-term technical data rights for weapon system sustainment. As part of the department’s acquisition reforms and performance-based strategies, the DoD has de-emphasized the acquisition of technical data rights. The GAO has recommended that the DoD recognize the need for the acquisition of technical data rights and asserts that without technical data rights, the DoD may face challenges in efficiently sustaining weapon systems throughout their lifecycle.

A very important contractual requirement for the prime contractor, General Dynamics Land Systems (GDLS), is to maintain an Operational Readiness Rate (ORR) of 90% or better. This requirement pertains only to the base vehicle configuration and does not include Government Furnished Equipment (GFE). Since initial deployment, Stryker has routinely exceeded this operational requirement. The Cost Plus Fixed Fee (CPFF) contract effectively motivates GDLS to exceed 90% ORR; however, the contract is not necessarily effective at controlling support costs, and this may be a risk to the government (US Army Audit Agency, 2005). One example of such a risk is the repair and replacement of a high-failure item—for example, cracked hydraulic reservoirs in the power pack. Maintenance procedures call for the entire power pack to be replaced as a unit, rather than removing and repairing/replacing the hydraulic reservoir within the power pack. Replacing entire power packs (instead of repairing/replacing hydraulic reservoirs within the power packs) results in shorter down-times and higher ORR, but it also requires more power packs (very large, expensive units) to be purchased and shipped to operating bases and forward maintenance facilities. The net result is that higher operational readiness is being purchased with increased transportation and storage costs.

Sustainability Challenges

Since Stryker's initial deployment was accelerated to meet an urgent combat need, the Stryker program team was performing the following activities concurrently: testing, production, fielding, training, and combat. In addition to the many challenges caused by these concurrent activities, the threat and operational environment in Iraq were different than anticipated, as previously mentioned. Several other factors added to the difficulty of maintaining Stryker vehicles in the field.

First, the Interactive Electronic Technical Manuals (IETMs) were not mature at the time of initial fielding. Many maintenance procedures could not be performed based on the IETMs because they were either not characterized correctly or crews were not adequately trained on their use. This situation led to *tribal system maintenance*, in which units depended on soldiers and contractors with experience on similar systems (like the M-113 armored personnel carrier) to figure out how to perform the maintenance actions correctly.

Second, since a large portion of maintenance actions were supported by contractor personnel, soldiers developed a *rental car mentality*. This lack of *ownership mentality* resulted in soldiers being overly dependent on contractor personnel to perform routine preventive maintenance actions, such as checking fluid levels. One vehicle was lost because the pre-mission engine oil check was ignored.

Findings

Stryker is performing well in the field. The system is exceeding expectations of Army management and soldiers. In spite of a changing threat environment (improved IEDs and excessive operations in the urban environment) and major configuration changes (5,000 pounds of add-on armor), Stryker is accomplishing its mission. The Operational Readiness Rate has consistently been over 90%.

Due to the increased threat of RPGs and IEDs, Stryker was outfitted with an add-on armor package. The additional 5,000 pounds of armor has been generally effective at mitigating the threat but has resulted in some negative operational/support consequences. The extra weight requires increased tire pressure, which causes operational problems and more structural stresses. Additionally, the armor limits crew visibility during operations and restricts airlift transportability on a C-130 aircraft.

Army decisions regarding contractor logistics support may remain with the Stryker program for years. When Stryker was first deployed to Iraq in 2003, the Army faced an immediate need for contractor maintenance personnel to support operations (45 vehicle maintenance personnel per brigade). The Army plans to eventually replace the 45 contractor maintenance personnel with soldiers, but it will take approximately 71 soldiers per brigade to perform the same level of vehicle maintenance as the 45 contractors because of other duties and responsibilities of active duty personnel. The current plan is to begin the transition to soldier maintenance in 2008, but the transition will probably be very difficult to implement due to the poor recruiting/retention outlook in general and to the shortage of appropriate active duty maintenance personnel.

Stryker program development was accelerated to meet the Army's combat needs in Operation Iraqi Freedom. Due to the compressed developmental schedule, Stryker DT/OT was unable to fully test all configuration changes. DT revealed relevant problem areas, but there was insufficient time or priority to correct all problems before OT and fielding.

For many DoD acquisition programs, the maturity of suitability parameters lags the maturity of effectiveness parameters during program development. Suitability determinants (such as reliability and maintainability) are not addressed early enough and are not prioritized with the same vigor and discipline as performance parameters like speed, accuracy, and lethality.

The general issue of suitability shortfalls in DoD acquisition programs is recognized at high levels of management and is being addressed. JROC, DOT&E, and USD(AT&L) have each called for increased attention to suitability improvements. For example, a new requirement exists for a Materiel Availability KPP.

The operational tempo of Stryker vehicles in Iraq far exceeds original usage estimates by at least 500%. Also, the mission profile of Stryker is much different than expected (80% on paved roads). This, in combination with the added weight of slat armor, has resulted in excessive stresses to the suspension, wheels, and tire assemblies, which causes increased failure rates of these items.

Since Stryker was fielded in 2003 in Iraq, the operational situation has been dynamic, unpredictable, and volatile. Four factors have made it very difficult to obtain complete and reliable data for trend analyses. The first factor is the rapidly evolving adaptive nature of the threat in an asymmetric combat environment. The second factor is that the operational environment for deployed Stryker vehicles is more severe than anticipated during design/development. The third factor is that, in response to the first two factors, configuration changes have precluded a stable baseline. The fourth factor is that in a dangerous combat scenario, recording and reporting data is not a high priority for operational crews.

Conclusions

In response to Operation Iraqi Freedom, there was an urgent operational need to deploy the Stryker system. Therefore, the development and test programs were greatly accelerated to get Stryker units into the field as quickly as possible. At the same time, the mission was changing as the threat quickly adapted and evolved in this asymmetric combat environment. The continually changing configuration baseline and changing tactical conditions made it very difficult to evaluate or predict reliability and suitability performance across all mission scenarios. The operational situation has been dynamic, as well as unpredictable and volatile, because Stryker was deployed in operational combat conditions that were different from, and much more complex than, those originally anticipated. In many ways, the system was not adequately designed for the actual threat it currently faces. However, this is certainly not the first time nor the last time this type of situation will occur. As a result, this case is a good example of how incomplete or incorrect maintenance/support planning can significantly add to the logistics burden. Due to the adaptive nature of the threat in the asymmetric warfare environment of Iraq and Afghanistan, our acquisition managers and operational planners are challenged to consider more complex and dynamic combat scenarios and contingencies than ever before.

List of References

Cloud, D. (2007, July 10). Army misses its June goal for new recruits. *New York Times*. Retrieved February 12, 2008, from www.nytimes.com/2007/07/10/washington/10military.html?ex=1341720000&en=9c451401c493106d&ei=5088&partner=rssnyt&emc=rss

Director, Operational Test and Evaluation (DOT&E). (2007, January). *FY-2006 annual report.* Washington, DC: Author.

Dougherty, J.E. (2004, March 3). *Stryker: Army's multimillion-dollar "lemon," or "excellent" lemonade?* Retrieved February 12, 2008, from http://archive.newsmax.com/archives/articles/2004/3/3/155315.shtml

Duma, W.D., & Krieg, K.J. (2005, June 20). *DoD guide for achieving reliability, availability, and maintainability*. Washington, DC: DoD.

GAO. (2004, August). *Military transformation: Fielding of Army's Stryker vehicles is well under way, but expectations for their transportability by C-130 aircraft need to be clarified* (GAO-04-925). Washington, DC: Author.

GAO. (2006, July). *Weapons acquisition: DoD should strengthen policies for assessing technical data needs to support weapon systems* (GAO-06-839). Washington, DC: Author.

GAO. (2006, September 5). *Defense logistics: Changes to Stryker vehicle maintenance support should identify strategies for addressing implementation challenges* (GAO-06-928R). Washington, DC: Author.

Joint Requirements Oversight Council. (2006, August 17). *Key performance parameter study recommendations and implementation* (JROCM161-06). Washington, DC: Author.

Levins, H. (2007, June 23). War exacts heavy toll on military vehicles, too. *St. Louis Post-Dispatch*, p. A2.

Military.com. (2007, July 2). *Hell on wheels: The Stryker combat vehicle*. Retrieved February 12, 2008, from www.military.com/soldiertech/0,14632,Soldiertech_Stryker,,00.html

National Research Council. (2004). *Improved operational testing and evaluation*. Washington, DC: The National Academies Press.

National Research Council. (2006). *Testing of defense systems in an evolutionary acquisition environment*. Washington, DC: The National Academies Press.

Smith, R.J. (2005, March 31). Study faults Army vehicle. *Washington Post*, p. A04.

Training and Doctrine Command (TRADOC). (2000). *Interim Armored Vehicle (IAV) operational mode summary/mission profile (OMS/MP)*. Fort Lewis, WA: Author.

Training and Doctrine Command (TRADOC). (2000, June 30). *Interim Brigade Combat Team (IBCT) organizational and operational concep*t (Ver. 4). Fort Lewis, WA: Author.

Tyson, A.S. (2003, October 9). New Army “Stryker” combat vehicle nears Iraq test. *The Christian Science Monitor*. Retrieved February 12, 2008, from, www.csmonitor.com/2003/1009/p02s02-usmi.html

US Army Audit Agency. (2005, December 6). *Stryker contract logistics support costs* (A-2006-0028-ALM). Alexandria, VA: Author.

US Army Test and Evaluation Command (ATEC). (2007, March 14). DoD/NDIA Suitability Conference. Hilton Head, SC: Author.

USD(AT&L). (2004, August 16). *Performance measurement metrics*. Washington, DC: Author.

USD(AT&L). (2006, November 23). *Total life cycle systems management (TLCSM) metrics*. Washington, DC: Author.

White, J. (2007, July 10). Army’s recruiting goal lags for several months in a row. *Washington Post*. Retrieved February 12, 2008, from http://globalpolicy.org/empire/challenges/overstretch/2007/0710recruitlag.htm

WorldNetDaily. (2003, June 4). *Billions wasted on new military vehicle?* Retrieved February 12, 2008, from www.worldnetdaily.com/news/article.asp?ARTICLE_ID=32898

Zagaroli, J. (2005, April 1). Report rips Army vehicle. *Detroit News*. Retrieved February 12, 2008, from www.globalsecurity.org/org/news/2005/050401-stryker-report.htm

THIS PAGE INTENTIONALLY LEFT BLANK

Panel 8 - Advances in Acquisition Project Management

Wednesday, May 14, 2008

3:30 p.m. – 5:00 p.m.

Panel 8 - Advances in Acquisition Project Management

Chair:

Joseph L. Yakovac, Jr., LTG, USA (Ret.), former Military Deputy to the Assistant Secretary of the Army for Acquisition, Logistics & Technology

Discussant:

Mr. Christopher A. Miller, Program Executive Officer Command, Control, Communications, Computers and Intelligence (C41)

Papers:

Capabilities Focused Acquisition Process

COL Raymond Jones, United States Army, Project Manager, Modular Brigade Enhancements

Defense Acquisition Management for Systems-of-systems

Dr. Daniel DeLaurentis, Assistant Professor, Purdue University

Chair: Lieutenant General Joseph L. Yakovac (Ret.) has served in a variety of command and staff positions at company grade through general officer ranks. He was a Platoon Leader, Executive Officer and Company Commander in mechanized infantry units. Following these assignments, Lieutenant General Yakovac earned a Master of Science degree in Mechanical Engineering from the University of Colorado. He subsequently served as an Assistant Professor at the United States Military Academy.

Yakovac's field grade assignments include Executive Officer and Branch Chief, Bradley Project Office; Brigade Logistics Officer, Brigade Operations Officer, and Battalion Executive Officer, US Army Europe; Staff Officer, Armor/Anti-Armor Task Force, Office of the Chief of Staff, Army; Mechanized Infantry Battalion Commander; Director, Weapons Systems Management Directorate, US Army Tank-automotive and Armaments Command; and Project Manager, Bradley Fighting Vehicle System.

Prior to Yakovac's position as the MILDEP, which he assumed in November 2003, he served most recently as the Program Executive Officer, Ground Combat and Support Systems (now known as Ground Combat Systems), and as Deputy for Systems Acquisition, US Army Tank-automotive and Armaments Command (TACOM). Before going to TACOM, his last position in the Pentagon was the Assistant Deputy for Systems Management and Horizontal Technology Integration, Office of the Assistant Secretary of the Army (Acquisition, Logistics and Technology).

Yakovac is a graduate of the Armor Officer Advanced Course, the Army Command and General Staff College, the Defense Systems Management College and the Industrial College of the Armed Forces. He wears the Expert Infantry Badge, the Ranger Tab, the Parachutist Badge, as well as the Distinguished Service Medal, three Legions of Merit and seven awards of the Army Meritorious Service Medal.

Discussant: Christopher A. Miller currently serves as the Program Executive Officer Command, Control, Communications, Computers and Intelligence (PEO C4I). In this capacity, he has oversight and responsibility for acquisition and lifecycle management for assigned C4I programs.

Miller, a native of Nashville, TN, received his Bachelor of Arts degree from Vanderbilt University through the Naval Reserve Officer Training Program. Upon commissioning in the United States Marine Corps and completion of The Basic School in Quantico, VA, Miller served as Intelligence Officer for various Marine Aviation Commands. In this capacity, he gained his experience in military intelligence and C4I leadership.

Miller left active duty status in 1999 to work for Booz Allen Hamilton in San Diego, CA. While a consultant, Miller worked on numerous command and control programs for the Navy and was integral in coordinating the Year 2000 transition for the Navy’s command and control programs.

In 2001, Miller returned to government service at the Space and Naval Warfare Systems Command (SPAWAR), where he provided technical and systems engineering leadership. He led the development of a common Windows 2000 PC software baseline to replace the then-current legacy Microsoft NT baseline. This software baseline is now the foundation of the Navy’s largest tactical network—known as Information Technology 21 (IT-21).

In 2004, Miller joined the PEO C4I staff and served in the positions of Technical Director and Director of Modernization. In these roles, he provided technical leadership and oversight for C4I program execution and fielding. Miller’s major accomplishments include leading a cross-service effort with the United States Air Force to establish guidance for implementing the Net-centric Enterprise Solutions for Interoperability (NESI), which was the PEO's first overarching guidance effort and a key enabler for delivering network-centric C4I capabilities. He also led the development of the first consolidated fielding plan and Modernization Concept of Operations (CONOPS), which defined and implemented the PEO’s modernization planning, design, and execution processes.

In May 2006, Miller entered the Senior Executive Service and was selected as Executive Director/Deputy for PEO C4I and Space. As Executive Director/Deputy, Miller led the program evaluation and integration efforts for all the Navy’s C4I acquisition programs of record. Miller was appointed Acting Program Executive Officer PEO C4I in August 2006.

Miller is a member of the Acquisition Corps, Armed Forces Communications and Electronics Association (AFCEA) and the United States Naval Institute (USNI). He has received several awards for his service, including a Navy and Marine Corps Achievement Medal and the AFCEA and USNI Copernicus Award.

Defense Acquisition Management for Systems-of-systems

Presenter: Shayani Ghose is a graduate student in the School of Aeronautics and Astronautics Engineering, Purdue University. She received her BS in Electrical Engineering and Computer Engineering (Dual) from Drexel University, Philadelphia, in June 2007. She is currently a part of the System-of-systems Research Group advised by Dr. Daniel DeLaurentis.

Shayani Ghose
Graduate Research Assistant, School of Aeronautics and Astronautics
Purdue University
701 W. Stadium Avenue
West Lafayette, IN 47907
Ph : +1 765 494 7958
Fax: +1 765 494 0307
Email: ghose@purdue.edu

Presenter: Daniel DeLaurentis is an Assistant Professor in the School of Aeronautics and Astronautics Engineering, Purdue University. He received his PhD from Georgia Institute of Technology in Aerospace Engineering in 1998. His current research interests are in Mathematical modeling and object-oriented frameworks for the design of system-of systems, especially those for which air vehicles are a main element and approaches for robust design, including robust control analogies and uncertainty modeling/management in multidisciplinary design.

Daniel DeLaurentis
Assistant Professor, School of Aeronautics and Astronautics
Purdue University
701 W. Stadium Avenue
West Lafayette, IN 47907
Ph : +1 765 494 0694
Fax: +1 765 494 0307
Email: ddelaure@purdue.edu

Abstract

The Department of Defense (DoD) has placed a growing emphasis in recent years on the pursuit of agile capabilities via net-centric operations. Dramatic technological advancements in communications and sensing have generated opportunities for battlefield systems to exploit collaboration for multiple effects. In this setting, systems are expected and often required to interoperate along several dimensions. Yet, the manner in which these “system-of-systems” are acquired (designed, developed, tested and fielded) has not kept pace with the shifts in operational doctrine. Systems acquisition remains largely focused on requirements for individual operation, paying insufficient attention to the ability of systems to influence the variety of future ecosystems in which they may subsist. Further, acquisition programs have struggled with complexities in both program management and engineering design. This paper establishes an understanding and classification of underlying complexities in the acquisition of system-of-systems. It also provides a conceptual model that exposes the connectivity between systems and the impact of system heterogeneity and externalities on that connectivity throughout the acquisition lifecycle. Implementation of this model in an exploratory simulation is in progress. Its objective is to allow acquisition professionals to develop intuition for procuring and deploying system-of-systems, providing a venue for experimentation and exploration to develop insights that underpin successful acquisition of SoS-oriented defense capabilities.

1. Introduction

A system-of-systems (SoS) consists of multiple, heterogeneous, distributed systems that can (and do) operate independently but can also assemble in networks and collaborate to achieve a goal. According to Maier (1998), component systems of the SoS typically demonstrate traits of operational and managerial independence, emergent behavior, evolutionary development and geographic distribution. Networks of component systems often form among a hierarchy of levels and evolve over time as systems are added to or removed from the SoS. However, these component systems are often developed out of context of their interactions with the future SoS. As a result, the systems may be unable to interact with the future SoS, adapt to any emergent behavior, or be robust in the face of external disturbances. The US Coast Guard's (USCG) Integrated Deepwater System (IDS) is an example of a Department of Homeland Security (DHS) acquisition process for an SoS, "patterned after the successful Department of Defense (DoD) model of contracting to competing industry teams" (Anderson, Burton, Palmquist, & Watson. 1999). IDS has faced technical and management challenges similar to those that are historically prevalent in acquisitions in SoS environments.

In the 1990s, the USCG Acquisition Directorate recognized the need to "deliver and support new generations of platform and mission systems" (1998). The 25-year, $24 billion IDS is aimed at "delivering new aircraft and cutters, modernizing legacy assets, and providing a new generation of Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) mission systems for forces deployed and ashore" (1998). In 2002, the Coast Guard awarded this contract to Integrated Coast Guard Systems (ICGS), an industry consortium of Northrop Grumman, Lockheed Martin and several other defense contractors. ICGS was contracted to act as the Lead System Integrator responsible for acquiring assets and integrating them into the IDS. The USCG recently dismissed its Lead System Integrator after a series of technical and managerial failures (Allen, 2007). In 2006, the Government Accountability Office (GAO) reported that the collaboration among the subcontractors continues to be problematic and that the system integrator wields little influence to compel decisions among them (Caldwell, 2006).

The DHS and the DoD are not the only organizations struggling with systems integration of a collection of complex system. The Air Transportation System and the NASA Constellation program are also facing similar challenges in attempting to apply generic system engineering processes for acquisition in an SoS environment. Integration challenges faced by the Constellation Program are documented in a recent NRC report (Committee on Systems Integration for Project Constellation, 2004). Both DoD and non-DoD examples are the key drivers motivating the research described in this paper.

The overarching goal of this research is to understand which types of acquisition management, policy insights and approaches can increase the success of an acquisition process in the SoS setting. The three research questions being explored are as follows:

1. Is there a taxonomy by which one can *detect* classes of complexities in particular SoS applications?
2. What are the underlying systems engineering (SE) and program management functions that are affected?

3. How can exploratory modeling generate SE and acquisition management insights and approaches to improve the probability of success?

In order to answer some of the questions posed, we aim to

1. Identify the complexities in the acquisition of SoS based on historical trends of “failures,” especially in the context of the DoD and DHS.
2. Develop a conceptual model of a generic acquisition process that can then be customized to different SoS applications.
3. Develop and simulate a computational model using an existing acquisition process for an SoS as a case-study; for example, the USCG Integrated Deepwater System could be used as an example of a DHS acquisition process set in an SoS environment. Interpretations of the results obtained would be used to field the research questions posed.

Since the project is presently at its midway point, in this paper we only focus on the first two research questions, specifically, on the mappings between SoS acquisition difficulties and complexities with a view toward model development. A general framework for and an outline of the computational model are provided.

2. Mapping Failure Modes to Underlying Complexities

Simon (1996) and Bar-Yam (2003) define complexity as the amount of information necessary to describe a system effectively. In the context of a system-of-systems, the necessary information encompasses both the systems that comprise the SoS and their time-varying interactions with each other and the “externalities.” Rouse suggested that the complexity of a system (or model of a system) is related to

- The intentions with which one addresses the systems.
- The characteristics of the representation that appropriately accounts for the system’s boundaries, architecture, interconnections and information flows.
- The multiple representations of a system, all of which are simplifications; hence, complexity is inevitably underestimated.
- The context, multiple stakeholders, and objectives associated with the system’s development, deployment and operation. (2007)
- (Polzer, DeLaurentis and Fry (2007) explored the issue of multiplicity of perspectives, in which perspective was defined as a system’s version of operational context.)
- The learning and adaptation exhibited during the system’s evolution (Rouse, 2007).

Historical data from previous unsuccessful defense acquisition programs show a distinct correlation with the causes for complexity identified by Rouse (2007). Fowler (1994) points out some of the causes for the failure of the Defense Acquisition Process to be “over specification and an overly rigid approach on development”: unreasonably detailed cost estimates of development and production, impractical schedules and extremely large bureaucratic overhead. Dr. Pedro Rustan, director of advanced systems and technology at the National Reconnaissance Office, identified four specific shortcomings in the acquisition process for defense space systems: “initial weapons performance requirements that are too detailed and lacking flexibility,” “insufficient flexibility in the budget process,” “a propensity to increase performance requirements in the middle of the acquisition cycle” and demands to field entirely new spacecraft to meet new requirement" (Spring, 2005). Riccioni (2005) used the United State Air Force (USAF) F-22 Raptor Program to illustrate shortcomings of the existing Defense Acquisition Process. Some of the recognized reasons for the failure of the F-22 Raptor Program were the ambitious nature of the set requirements, the "gross underestimation" and continual misrepresentation of cost in order to "seduce the Congress and the Public" to believe that the aircraft was affordable. However, the major failing of this program lies in that the enormous delay in the development process (spanning over two decades) resulted in an aircraft that was no longer needed since the "existing and future enemies changed natures." Riccioni also points out that "terrorists are the only extant and foreseeable threats" but they do not threaten the West with fighter aircrafts. In another example, the US Army's Future Combat Systems (FCS) has found difficulties in developing and fielding equipment that meet the program objective (Capaccio, 2006).

Using the above examples in conjunction with other acquisition programs, such as USCG Integrated Deepwater System and Future Combat System, we summarize the common causes of failure within acquisition processes as: a) *misalignment* of objectives among the systems, b) limited *span of control* of the SoS engineer on the component systems of the SoS, c) *evolution* of the SoS, d) *inflexibility* of the component system designs, e) *emergent behavior* revealing hidden dependencies within systems, f) *perceived complexity* of systems and g) the challenges in *system representation*.

Sage and Biemer (2007) examined the existing systems engineering process models in the context of their applicability to SoS and concluded that none of them "could be tailored to systems family development." Sage and Biemer also developed the System-of-systems Engineering (SoSE) Process Model designed specifically for SoS applications. The complexities discussed above were mapped onto a section of the SoSE Process Model based on the trends observed in past acquisition processes within the DoD Acquisition Process. Figure 1 depicts the mapping of some of these complexities to a section of the SoSE Process Model, representing from where complexities might arise and how they may affect the acquisition process. For example, SoS operations could demonstrate emergent behavior and result in a change in the CONOPs for the SoS. Evolution of the SoS changes the CONOPS of the SoS may result in a subsequent change in the Acquisition Strategy. Misalignment of objectives of the component systems in an SoS can arise from both the CONOPs as well as the SoS Project Control. System inflexibility, perceived complexities and challenges in representing systems occur mostly between or within systems. Accurate representation of component systems is complicated by the presence of hidden and visible dependencies between systems, fuzzy boundaries, unknown architectures, etc.

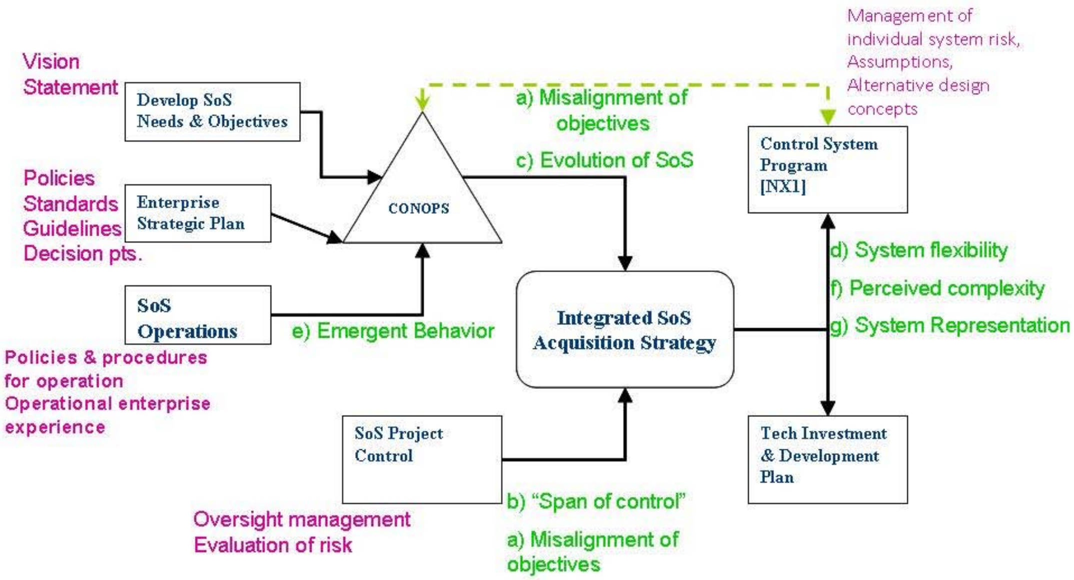


Figure 1. Complexities Mapped to a Section of the SoSE Process Model
(Sage & Biemer, 2007)

3. Towards Development of an Exploratory Model for SoS Acquisition

3.1 Pre-Acquisition Model

The purpose of the pre-acquisition model is to better understand the external stakeholders that affect the acquisition process. The model we developed is depicted in Figure 2 and is based loosely on Sage and Biemer (2007) SoSE Process Model. External inputs to the SoS acquisition process are sorted into three categories: "Capabilities & Possibilities" (CAP), "Technology Assessment, Development, Investment and Affordability Plan" (ADIA) and the funding received. The CAP and ADIA are our own creation. Though they are similar to the *Concepts of Operation* (CONOPs) and *Technology Investment and Development Plan* (TDIP) in the SoSE Process Model, there are some key differences.

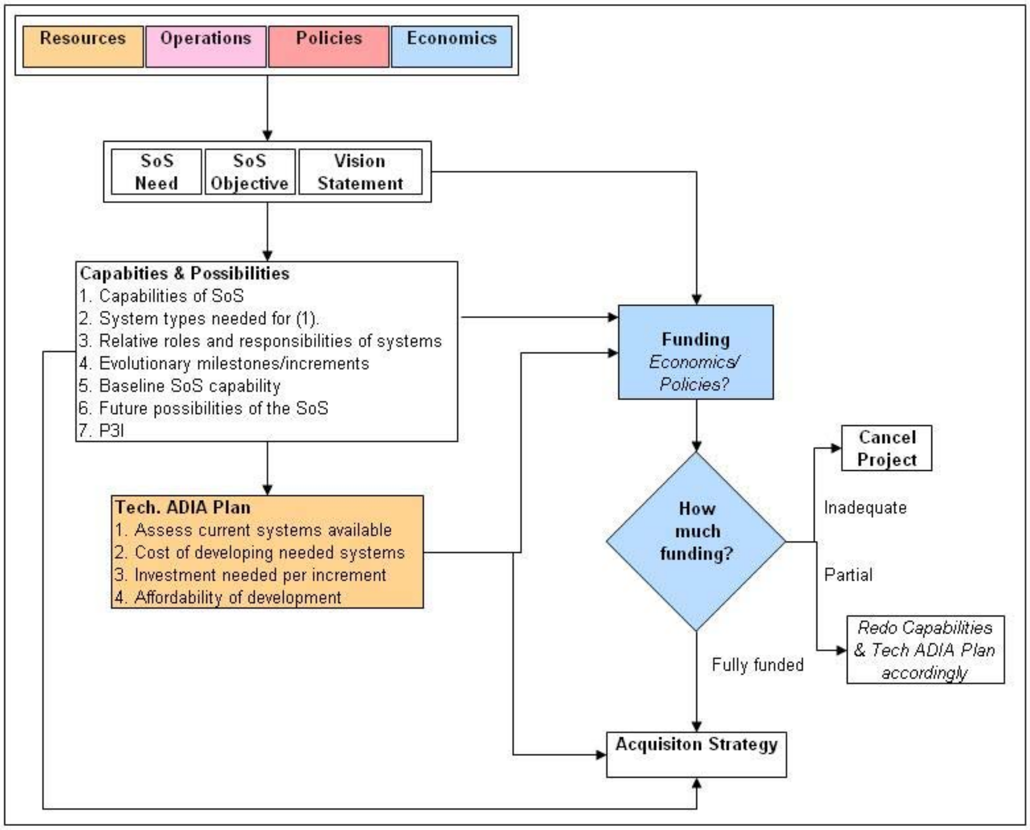

Resources
Operations
Policies
Economics
SoS Need
SoS Objective
Vision Statement
Capabities & Possibilities
1. Capabilities of SoS
2. System types needed for (1).
3. Relative roles and responsibilities of systems
4. Evolutionary milestones/increments
5. Baseline SoS capability
6. Future possibilities of the SoS
7. P3I
Tech. ADIA Plan
1. Assess current systems available
2. Cost of developing needed systems
3. Investment needed per increment
4. Affordability of development
Funding
Economics/
Policies?
How much funding?
Cancel Project
Inadequate
Partial
Redo Capabilities & Tech ADIA Plan accordingly
Fully funded
Acquisiton Strategy

Figure 2. Conceptual Model for Pre-Acquisition Activities

The need, objective, and vision for an SoS feed the CAP. The CAP is a high-level requirements document that provides the following information:

1. The capabilities that the SoS is required to possess and services it must provide
2. The system types that are needed to provide these capabilities
3. The relative roles and responsibilities of the constituent systems
4. Milestones in the development of the SoS and the number of increments needed
5. Baseline SoS capability at its first deployment
6. Future possibilities for the SoS in terms of capabilities it may possess and services it may provide
7. Pre-planned product improvement (P3I) for each system type to support future capabilities of the SoS

The main differences between the CONOPS and CAP stem from the last two entries (6 and 7). The evolutionary nature of the SoS requires the dynamic addition and removal of component systems and functionalities. While the capabilities required for the SoS at the time of first deployment may be basic, the future capabilities of the SoS allow the systems of the SoS to be developed keeping in mind the future capabilities the SoS will provide. This prevents individual systems from becoming obsolete or being drastically re-designed.

The CAP feeds the Technology ADIA Plan, which is needed to

1. Assess the capabilities of the legacy systems and their current maturities
2. Provide a cost estimate for upgrading/ developing systems
3. Provide an estimate of the investment that will be needed per increment
4. Assess the affordability of the investment

The differences in the Technology ADIA Plan and the TDIP stem from entries 2 and 3 in the list above. In addition to the functions of the TDIP, the Technology ADIA Plan provides cost estimates for all new development needed, including a cost breakdown per increment in the development process.

Both the CAP and the Technology ADIA Plan are required to determine the amount of funding required for the project. While there are numerous factors that are used to determine funding (such as political affiliation, unexpected crisis, regulations etc), CAP and the Technology ADIA Plan are the inputs to the acquisition process that translate into technical requirements for the SoS. Provision of a computational model of the pre-acquisition activities is outside the scope of this paper. Instead, we focus on realizing a model for the acquisition strategy, which is described next.

3.2 Acquisition Strategy Model

Development of a “brand new” SoS has been and will remain a rare occurrence. The United States Air Force (USAF) Scientific Advisory Board (Saunders et al., 2005) states that one of the challenges in building an SoS is that legacy systems are contributors that affect the performance of other systems. These legacy systems, may be used “as-is” or may need some re-engineering to feed the needs of the new SoS. In addition, new systems are incorporated to develop the capabilities of the SoS. Again, the new systems can range from off-the-shelf, plug-and-play products to custom-built systems dependent of the working of a legacy system. The breadth of the heterogeneity of the components can be broadly categorized under legacy systems, new systems and improvements. Sub-categories arise when two or more categories overlap.

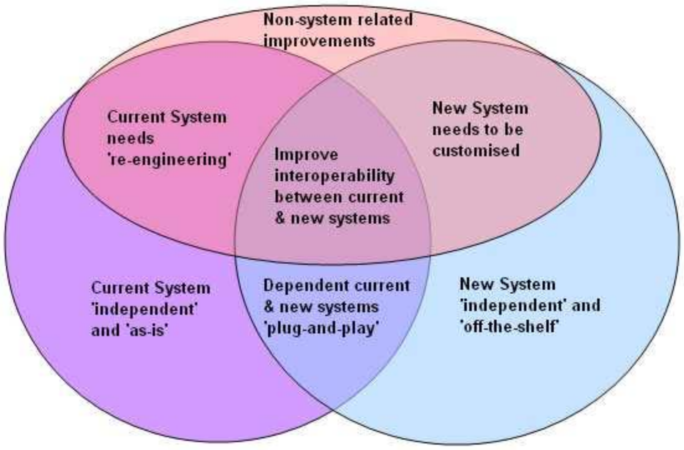

Non-system related improvements
Current System needs 're-engineering'
Improve interoperability between current & new systems
New System needs to be customised
Current System 'independent' and 'as-is'
Dependent current & new systems 'plug-and-play'
New System 'independent' and 'off-the-shelf'

Figure 3. Heterogeneity of Component Systems in an SoS

The different components that comprise the acquisition process in an SoS environment are depicted in Figure 3. For example, improvements can be non-system related, such as improvements in business practices for the SoS, or they can be system-related such as re-engineering legacy systems or customizing/developing new systems to meet the needs of the SoS. Similarly, legacy and new systems can be independent “as-is” systems, dependent “as-is” systems, independent systems in need of “re-engineering,” or dependent systems that need customization. Another subcategory is based on the interoperability of the systems in the SoS. While some dependent systems have existing interfaces that allow them to be interoperable (plug-and-play), others need to develop interfaces that allow them to interact with other systems.

Implementing and integrating these different kinds of systems and processes into the SoS is made more complex by the evolutionary nature of an SoS. Thus, it must be made possible for component systems to re-designed or upgraded dynamically without having to re-design the entire SoS. Also, though most systems depend on others during the implementation or integration phases, they are not centrally controlled. This requires that the systems have an incentive to collaborate with each other without being forced to do so. These issues are merely a sub-set of the challenges for an acquisition process (as discussed in Section 2) in an SoS environment.

The conceptual model for acquisition strategy proposed in this section is based on the 16 basic technical management and technical system-engineering processes outlined in the Defense Acquisition Guidebook (DoD, 2006), often referred to as the 5000-series guide. However, an SoS environment changes the way these processes are applied. The 2007 System-of-Systems System Engineering (SoS-SE) Guide (Systems and Software Engineering, 2006) addresses these considerations by modifying (or in some cases revamping) some of the 16 processes in accord with an SoS environment. These new processes and their functions are described in Table 1. Our conceptual model for acquisition in an SoS environment is illustrated in Figure 4. It is centered on the revised processes and depicted in a hierarchy to show the flow of control between the processes throughout the acquisition lifecycle.

Table 1. Modified Technical Management and Technical Processes as described in the DoD SoS-SE Guidebook
(Systems and Software Engineering, 2006)

Requirements Development	Takes all inputs from relevant stakeholders and translates the inputs into technical requirements
Logical Analysis	Is the process of obtaining sets of logical solutions to improve the understanding of the defined requirements and the relationships among the requirements (e.g., functional, behavioral, temporal)
Design Solution	Process that translates the outputs of the Requirements Development and Logical Analysis processes into alternative design solutions and selects a final design solution.
Decision Analysis	Provides the basis for evaluating and selecting alternatives when decisions need to be made.
Implementation	The process that actually yields the lowest level system elements in the system hierarchy. The system element is made, bought or reused.
Integration	The process of incorporating the lower-level system elements into a high-level system element in the physical architecture.

Verification	Confirms that the system element meets the design-to or build-to specifications. It answers the question, “Did you build it right?”
Validation	Answers the question “Did you build the right thing?”
Transition	The process applied to move the end-item system to the user.
Technical Planning	Ensure that the systems engineering processes are applied properly throughout a system’s lifecycle.
Technical Assessment	Activities measure technical progress and the effectiveness of plans and requirements.
Requirements Management	Provides traceability back to user-defined capabilities
Risk Management	To help ensure program cost, schedule and performance objectives are achieved at every stage in the lifecycle and to communicate to all stakeholders the process for uncovering, determining the scope of, and managing program uncertainties.
Configuration Management	The application of sound business practices to establish and maintain consistency of a product’s attributes with its requirements and product configuration information.
Data Management	Address the handling of information necessary for or associated with product development and sustainment.
Interface Management	Ensures interface definition and compliance among the elements that compose the system, as well as with other systems with which the system or systems elements must interoperate.

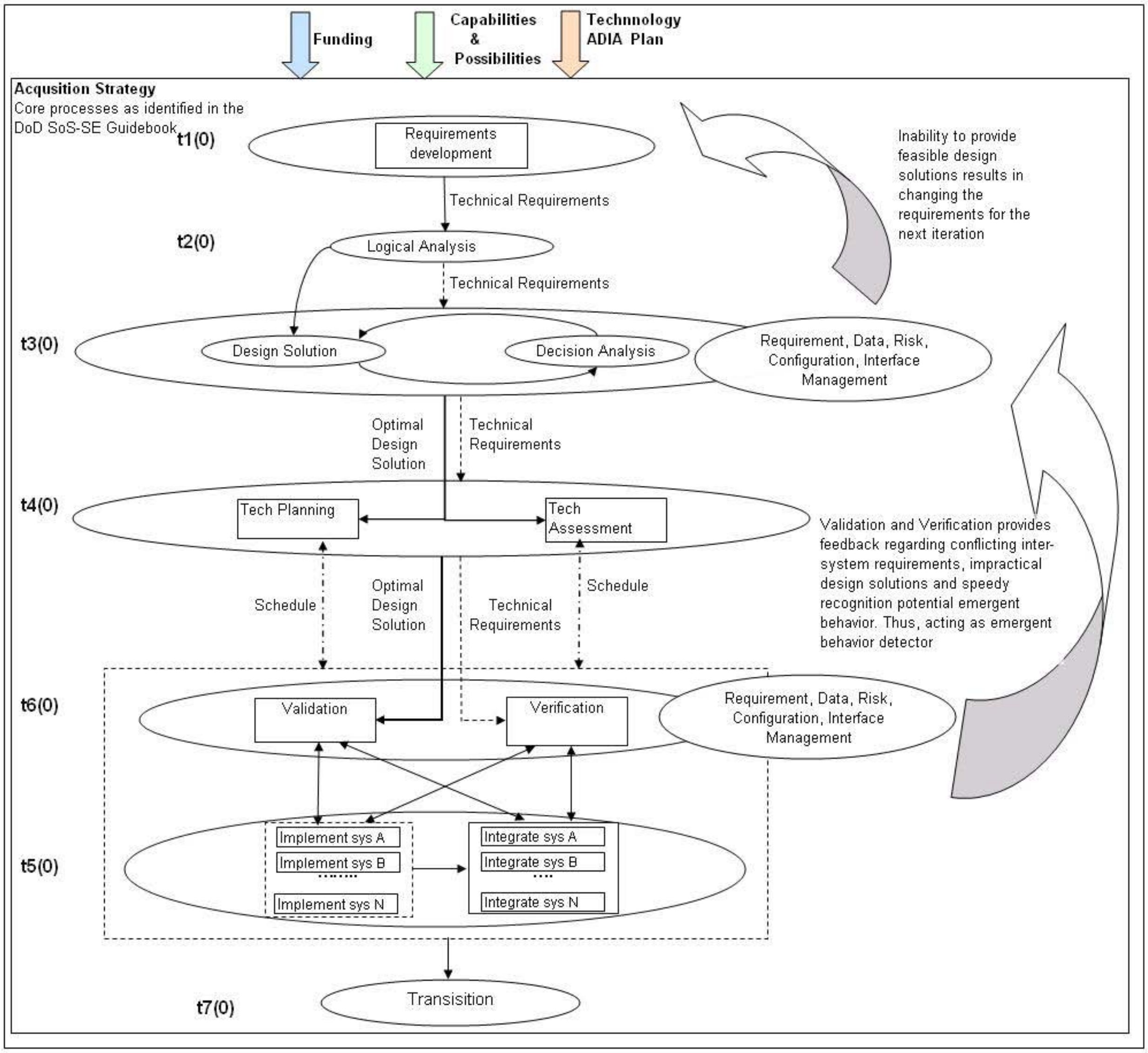

Funding
Capabilities & Possibilities
Technnology ADIA Plan
Acqusition Strategy
Core processes as identified in the DoD SoS-SE Guidebook
t1(0)
Requirements development
Inability to provide feasible design solutions results in changing the requirements for the next iteration
Technical Requirements
t2(0)
Logical Analysis
Technical Requirements
t3(0)
Design Solution
Decision Analysis
Requirement, Data, Risk, Configuration, Interface Management
Optimal Design Solution
Technical Requirements
t4(0)
Tech Planning
Tech Assessment
Validation and Verification provides feedback regarding conflicting inter-system requirements, impractical design solutions and speedy recognition potential emergent behavior. Thus, acting as emergent behavior detector
Schedule
Optimal Design Solution
Technical Requirements
Schedule
t6(0)
Validation
Verification
Requirement, Data, Risk, Configuration, Interface Management
t5(0)
Implement sys A
Implement sys B
Implement sys N
Integrate sys A
Integrate sys B
Integrate sys N
t7(0)
Transisition

Figure 4. Conceptual Model for Acquisition Strategy Based on SoS SE Processes of Table 1

As can be seen in Figure 4, *Requirements Development* provides the technical requirements of the SoS, based on the relevant external inputs. The pre-acquisition model as discussed in Section 3.1 provides details about the external inputs: CAP, Technology ADIA Plan and funding. The technical requirements are then sent to *Logical Analysis* to check for relationships among the requirements. This also helps to check for inconsistencies among requirements for different systems and how that might affect the functioning and behavior of the future SoS.

Design Solution development and *Decision Analysis* are the next processes that come up with the optimal design solution from the set of feasible solutions to meet the given requirements. The optimal design solution is not only based on the current set of requirements and solution alternatives but it also takes into account all previous information and data available through requirements, risk, configuration, interface and data management processes. Since most SoS acquisitions are multi-year projects involving many different parties, the overlap between the management processes, *Design Solution* and *Decision Analysis* processes, allows for greater traceability for decisions made. The optimal design solution obtained from this phase is then sent to the next stage: *Technology Planning* and *Technology Assessment*. In the event that there is not an optimal or sub-optimal design solution to successfully implement the given requirements, the feedback loop to *Requirement Development* translates into a change in the technical requirements for the SoS.

Technology Planning and *Technology Assessment* are essentially scheduling processes that help oversee the implementation, integration, validation and verification for all the α-level systems in the SoS. Systems in the SoS are often dependent on other systems for either implementation or integration, or both. These dependencies correspond to time-lags in the acquisition process. For example, if system A is a legacy system and system B is being built, the integration of A with B will not occur until B has been completely implemented. This generates a time lag, especially if another system C is waiting to be implemented based on the integration of A with B. As more systems are added to the SoS, it becomes necessary to generate a schedule that can help coordinate the process. This schedule also needs to be continually updated to reflect unexpected delays, clearly identify bottle-necks, etc.

Due to the heterogeneity of the systems that comprise the SoS and the interactions between then, *Validation* and *Verification* processes need to not only check for suitable implementation of the “optimal design solution” on a system-level but also need to be on the lookout for any misaligned objectives between systems, hidden dependencies among the systems, and any emergent behavior that may affect the functioning and/or behavior of the future SoS. In most situations, early detection of an emergent behavior will prevent the re-designing of major system components and ensure that the SoS functions satisfactorily. Even though *Validation* and *Verification* processes oversee *Implementation* and *Integration*, they occur after *Implementation* and *Integration* have begun.

While *Implementation* and *Integration* are the lowest levels of the acquisition model shown, much of the feedback from this level translates into developing different design solutions and sometimes changing the technical requirements. This level deals with acquiring the systems in the SoS and integrating them based on their dependencies with other systems. These processes consume the bulk of the financial and other resources as well as consume the most time. Therefore, it is understandable why system engineers are often reluctant to re-design functional systems on a whim and why they want to make sure that once the system has been developed, integrated and tested, it does not go back into the *Implementation* phase.

4. Developing a Computational Exploratory Model for Acquisition Strategy

4.1 Overview

Our purpose in constructing a computational exploratory model is to allow acquisition professionals to develop intuition for procuring and deploying system-of-systems. Thus, the objective is not to provide a model validated and ready for deployment in real acquisition programs but to expose the complexities in SoS acquisition. The specific complexities targeted are related to evolutionary development of the SoS and the span-of-control possessed by the SoS managers and engineers. The conceptual model displayed in Figure 4 is implemented using the USCG Integrated Deepwater System as a case study. Given the possibility of emergent behavior during evolutionary development and the heterogeneity of the components and their interactions, we decided to use the agent based modeling (ABM) approach.

Several challenges arise in transforming the acquisition model to a computational model for purposes of simulation and learning. One challenge lies in converting all the qualitative concepts into quantitative measures to support the computational model for SoS acquisition. We started by building an ideal model devoid of disruptors that historically plague the acquisition process. We will add non-linear behavior to the ideal model to test different scenarios in the process.

A second challenge is building a model that can accommodate the dynamic addition and removal of components in the SoS. In addition, these component systems need to reflect the heterogeneity of the systems in a real acquisition process. We included parameters such as *level of completeness* to demonstrate the difference between legacy systems, new systems and the partially implemented/ integrated systems. *Level of completeness* for both integration and implementation processes vary between 0 and 1, where 1 means that the system is 100% complete. For example, system A representing a fully implemented legacy system has an implementation completeness set to “1,” while its integration with system B might only be 20% complete. In this case, the completeness of the integration phase is initialized to 0.20.

A third challenge arises from the numerous methodologies that can be applied to reflect the integration and implementation processes. In a simplified model, it is much easier to begin integration once all the systems have been implemented. However, this method is neither cost nor time efficient, especially in multi-year projects involving numerous systems. On the other hand, dynamically implementing and integrating systems is time efficient but often not possible when dependent systems are outside the span of control of the systems engineers. For example, a cutter and helicopter may be dependent systems being developed by different contractors or different groups that cannot be forced to collaborate. This gives rise to questions: How do we group the systems for integration to achieve maximum efficiency with regard to time and cost incurred? Would it be beneficial to group systems based on the span of control or influence of the systems engineers or on the direct dependencies of the system?

However, developing an acquisition model that studies the effects of all the factors that add to the complexity of the acquisition process for SoS in a short span of time is impossible. Our coarse-scale engineering model will specifically target challenges related to the evolution of the SoS and the span-of-control of the SoS engineer(s).

4.2 Model Inputs

The exploratory computational model developed has a top-down flow with feedback at two junctures. The flow of control begins from Requirement Development (Level t0(0), Figure 4) to Design Solution (Level t3(0), Figure 4) through Logical Analysis(Level t2(0), Figure 4). This linkage is shown in the Figure 5.

The primary inputs fed into the Requirement Development and Logical Analysis processes (e.g., number of requirements, the relationships between these requirements, the systems affected and the dependencies between these systems) are user-defined. The inputs are used to develop a schedule of when each requirement will be implemented, depending on the relationships between the requirements. An example of such user-defined data for these steps is shown in Table 2.

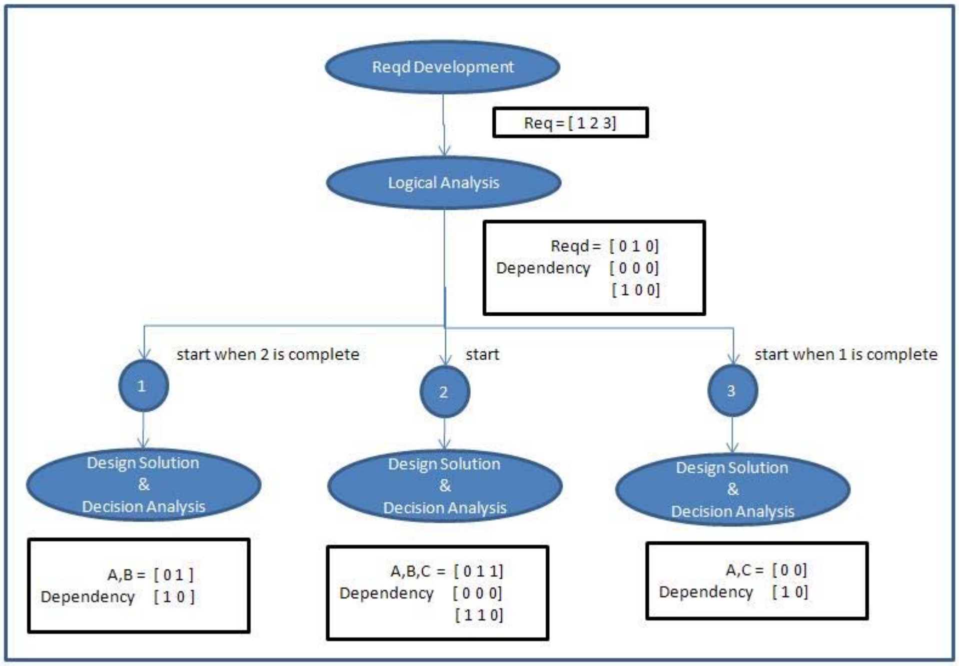


Figure 5. Flow of Control and Parallel Processing of Requirements

As shown in Table 2, there are 3 requirements (1, 2 and 3) and each has a dependency vector associated with it. The vectors are concatenated to form the dependency matrix for requirements. An "X" is placed for all diagonal elements of the dependency matrix because a requirement cannot be dependent on itself. The vector for requirement 1 ([X 1 0]), shows that requirement "1" is dependent on requirement "2." This means that requirement 1 cannot be realized until requirement 2 is implemented. A lack of dependency between requirements means that the requirements can be simultaneously realized. In real world applications, communication upgrade to the North-Atlantic fleet may be independent of the weaponry upgrade for the same group of systems. In such a case, both the requirements on the same group of systems may be implemented simultaneously. Each requirement affects a subset of the systems present in the SoS, and the systems in each subset share a unique dependency matrix with other systems in that subset (shown in Table 2).

Table 2. User-defined Data Used for the Computational Model

Requirements	Dependency	Systems Affected	System-level Dependency
1	[X 1 0]	[A, B]	$\begin{bmatrix} X & 1 \\ 1 & X \end{bmatrix}$
2	[0 X 0]	[A, B, C]	$\begin{bmatrix} X & 1 & 1 \\ 0 & X & 0 \\ 1 & 1 & X \end{bmatrix}$
3	[1 0 X]	[A,C]	$\begin{bmatrix} X & 0 \\ 1 & X \end{bmatrix}$

All component systems of the SoS have user-defined and calculated parameters that expose their heterogeneity and help track their progress through the acquisition process. Some of the parameters used to describe each system in the SoS are described in Table 3. While the parameters “Name,” “Imp.completeness,” “Imp.time,” “Imp.dependencies,” “Int.completeness,” “Int.time” and “Int.dependencies” are user-defined, ID and Mode are calculated by the model.

Level of completeness for both integration and implementation processes vary between 0 and 1, where 1 means that the system is a 100% complete. A partially complete system may start with a fractional level of completeness. Though the *level of completeness* for implementation and integration are mutually exclusive, a system can not have “0” implementation completeness and a non-zero integration completeness. This qualitatively means that a system that has not yet been implemented cannot be partially integrated with another system. Times to complete implementation and integration are discrete.

Table 3. Parameters Used to Describe Any System in the SoS

Parameter	Description
Name	Name of the system
ID	Unique ID assigned to the system
Imp.completeness[]	An array that gives the completeness of the implementation of the system at any given time.
Imp.dependencies[]	Dependency vector that shows if system implementation is dependant on any other system
Imp.time	Maximum time needed to complete implementation
Int.completeness[]	An array that gives the completeness of the integration of the system with respect to another system at any given time.
Int.dependencies[]	Dependency vector that shows if system integration is dependant on any other system
Int.time	Maximum Time needed to complete integration of system x with any other system
Mode[x;y]	Provides the phase of development the system is currently in and its status

Each system has a pre-defined dependency vector associated with implementation and integration processes. These vectors are concatenated to form a dependency matrix for the systems affected by each requirement. Elements along the diagonal of the dependency matrix (denoted with an “X”) are assigned a value of “0” since a system cannot be dependent or independent of itself. Otherwise, element (i,j) in the system-level dependency matrix can be 0 or 1. A value of 0 signifies that the system (i) is independent of system(j) and a value of 1 signifies that system(i) is dependent on system(j). The dependency matrix can be directed. This occurs when system (i) is dependent on system (j) but not vice-versa. In Table 3, the system-level dependency matrix for requirement 3 is directed.

As previously mentioned, parameters of ID and Mode are assigned by the model. When the system is added to the SoS, it is assigned an ID to uniquely identify it throughout the lifecycle of the SoS. The mode of each system contains two elements: the phase (first element) and the status (second element). Depending on the level of completeness of the system in the integration and implementation phases, the phase of the mode is set to values of {0,1,2,3}. When the system is added into the SoS, 0 is the Mode assigned; 1 and 2 refer to implementation and integration states respectively, and 3 refers to in-operation state. Status of the Mode can be set to 0 or 1, depending on if the system is available for implementation/integration or not. Thus, a Mode of [2; 1] means that the system is in the integration phase and is currently being integrated with another system.

4.3 Model Dynamics

As seen in Figure 5, the flow of control in the model starts from the *Requirement Development* (Level t0(0), Figure 4) to *Validation and Verification* (Levelt6(0), Figure 4). When the model is first deployed, this stage initializes all processes by supplying requirements to be implemented, systems affected, etc. Each requirement also has a “change” status flag that shows when a particular requirement is unchanged (status=0), being changed (status=-1) or has been changed (status=1). When a requirement is changed after the acquisition process has begun, it affects all subsequent processes.

Using the user-defined parameters and inputs from *Requirement Development* (similar to the data shown in Table 2), *Logical Analysis* (Level t2(0), Figure 4*)* generates a schedule to realize the given requirements. Depending on the dependencies between the requirements, they get implemented in series or in parallel with other requirements. As shown in Figure 5, every requirement being implemented is fed into its own *Design Solution* and *Decision Analysis* (Level t3(0), Figure 4*)* process. This process can change the implementation or integration times required by each system affected by the requirement. If a requirement is changed, the design solution and implementation processes for the component systems are affected. The set of systems with their modified implementation and integration times are then sent through the *Technology Planning* and *Technology Assessment* (Level t4(0), Figure 4) processes.

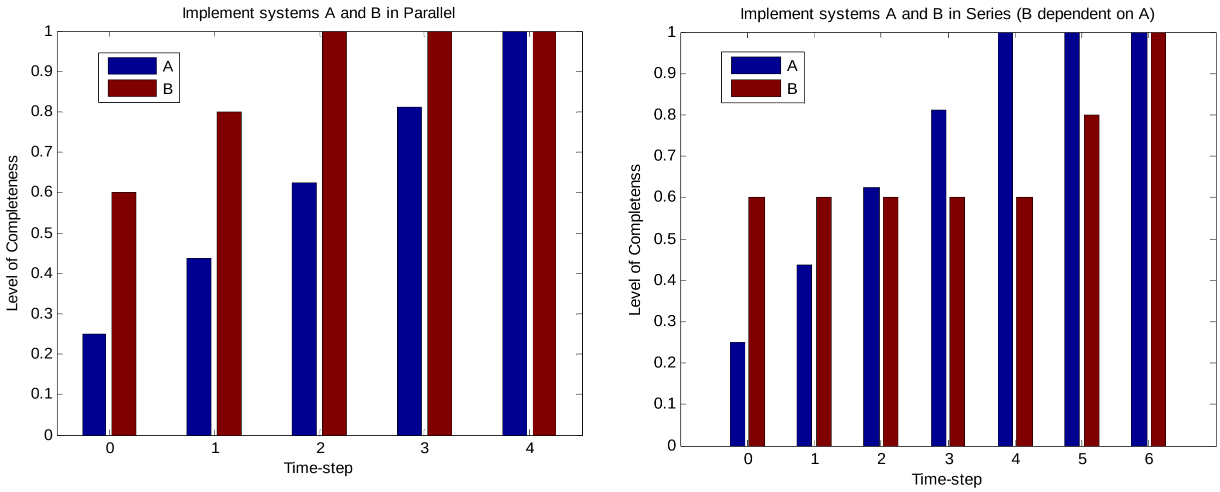
Technology Planning takes in the array of systems being affected by the given requirement and divides them into “to-be-implemented” and “read-for-integration” queues depending on the implementation/integration times needed for each system. Since the component systems in the acquisition process can be at varying levels of completeness during the implementation and integration cycles, they are dynamically added to the queues. For each queue, a synchronization matrix (sync_mat) is generated to keep track of the number of systems in the queue, their expected times of completion and their “iteration-rate.” Given the maximum time allotted and the existing level of completeness, Iteration rate is defined as the average rate at which the process needs to be completed for a system. For example, if system "A," which is 25% completed needs to be fully implemented in 5 time-steps, using Equation 1, the iteration rate of system "A" is calculated to be 0.15.

Equation 1. $$Iteration\_Rate = \frac{1 - completeness(0)}{\max\_time}$$

The systems in the synchronization matrix for the implementation/integration queues are sorted on the basis of 1) number of systems dependent on that system and, 2) maximum time required by the system to complete the process. Thus, systems with larger number of systems dependent on them are higher on the synchronization queue than those with a fewer number of dependent systems. Similarly, systems with the same number of dependent systems—those requiring less time for implementation/integration—get higher priority. Since this is a basic model, we only have two criteria to determine an appropriate schedule. In future models, more conditions can be added to simulate different methodologies. For example, priority of the systems in the queue may be a factor that determines their position in the process queues.

Technology Assessment uses the synchronization matrix developed by *Technology Planning* to track the progress of the systems in the implementation/ integration queues. In a non-ideal acquisition model, the iteration rates for the systems will be subject to external perturbations. A drastically reduced iteration-rate will stall the development of a system mid-process and affect other systems dependent on the stalled system. *Technology Assessment* recognizes the stalled systems and activates "enablers" to re-adjust their iteration-rate.

Implementation (Level t5(0), Figure 4) of systems can also occur in series or parallel configurations depending on the system dependencies. The level of completeness for an implementation process increases by the iteration rate at every time-step, until it reaches a completeness value of 1. The incremental increase in the level of completeness of two independent systems with different iteration rates occurs simultaneously, as shown in Figure 6a. If system "B" were dependent on system "A," then implementation of B would commence when A was fully implemented, as shown in Figure 6b. When a system achieves the implementation completeness value of 1, it is added onto the integration queue. Since the integration and implementation process queues are dynamically generated, the synchronization matrix for the systems in the queues also changes dynamically.



a) Independent Systems

b) Dependent Systems

Figure 6. Incremental Increase in Implementation Completeness for a) Independent Systems b) Dependent Systems.

The *Integration* (Level t5(0), Figure 4) process also uses a synchronization matrix for coordination. While parallel processing of dynamically changing process queues greatly improves the time efficiency for independent systems, they also increase the probability of dependent systems trying to integrate with each other at the same time. For example, system "A" might be waiting to be integrated with system "B" while system "B" is being integrated with system "C." The current implementation algorithm solves this problem by using the Mode parameter for each system. As described in Table 3, Mode is a 2x1 array structure. The first element gives the phase the system is currently in and the second element gives the status. When a system in the integration queue (phase 2) starts getting integrated with another system its status changes from 0 to 1. In the previous example, when "B" begins integration with system "C," the mode status of "B" changes to 1. System "A" cannot integrate with system "B," unless the mode status for "B" changes back to 0. This process is better illustrated in Figure 7.

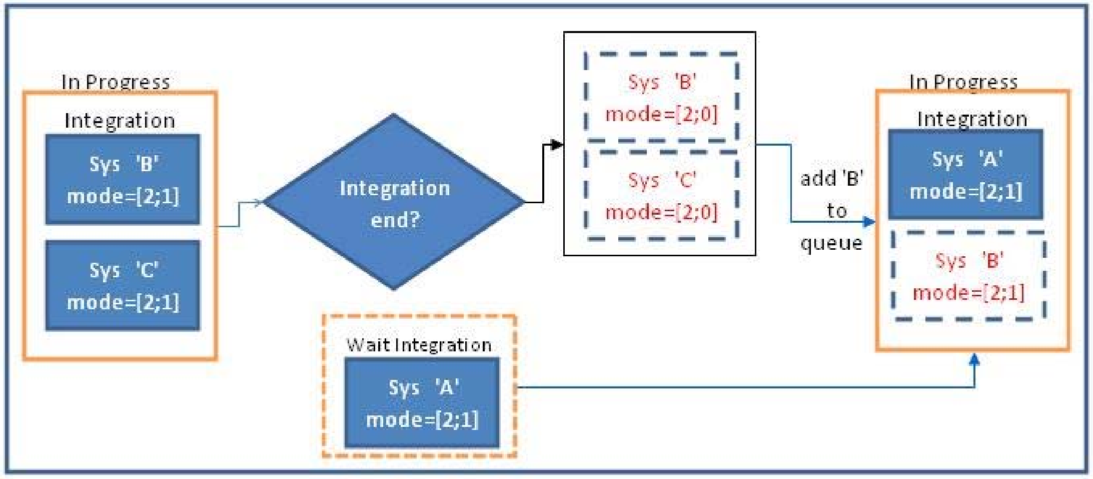


Figure 7. Use of Mode in the Integration Process Queue

When both the *Implementation* and *Integration* processes for all the systems affected are complete, the *Validation* and *Verification* (Level t6(0), Figure 4) processes check to see that all the systems were implemented and integrated within constraints of time and budget. They check for a completeness level of 1 for all the component systems affected, and then they compare the actual time needed and cost incurred by each system to the time and cost allotted to them by *Design Solution* and *Decision Analysis*.

5. Work in Progress: Implementation and Success Metrics

As discussed in Section 4, the modeling approach for acquisition allows for a great deal of flexibility in terms of parameters, methodologies, etc. While we have generated a successful "small-scale" ideal model to depict the basic methodology of the defense acquisition process, we have yet to implement different scenarios using disruptors and enablers.

Also, while the feedback system within the system-level has been implemented between processes like integration and technology planning, the feedback system between *Design Solution* and *Requirements Development* has yet to be implemented. The high-level feedback loops to *Requirements Development* and *Logical Analysis* will allow us to model scenarios in which requirements change at different times through out the lifecycle of the SoS. As mentioned in Section 2, historical trends in defense and non-defense related acquisition processes show that requirement changes occurring during the later phases of the acquisition lifecycle have been major contributors to the failure of the SoS acquisition program.

We are also currently working on developing the *soft* parameters that allow for fuzzy boundaries depicting varying spans of control of the SoS engineers and managers over the different component systems This may be reflected in the system-dependency matrix by using values between 0 and 1. Fractional values of dependency then need to be mapped to a time-delay parameter in each process, in order to show the relationship (if any) between the span of authority of a systems or system-of-systems engineer and the time required for the completion of an acquisition process.

The success of the developed ABM Model will be measured by the ability of the model to provide insights into the types of acquisition management insights and approaches that can increase the success of an acquisition process in the SoS setting. The model will enable us to answer the three research questions posed in Section 1, specifically targeting complexities related to span of control/influence of the SoS engineers and managers and the evolutionary development of an SoS. A successful model will allow for the study of various scenarios generated by implementing different acquisition management strategies and approaches in an SoS environment. For example, scenarios could be generated by adjusting the "levels" of collaboration between individual system engineers at varying hierarchical levels or the span of control of the SoS engineer throughout the lifecycle of the SoS. The model may even be used as a comparative tool to study the effect of implementing different methodologies for individual processes on SoS parameters, such as time needed to acquire, test and deploy specific capabilities. The greatest benefit of such modeling lies in its ability to act as a decision-making tool for SoS engineers, program managers and systems engineers to improve the probability of success of the SoS acquisition process by simulating different scenarios and implementing different strategies.

6. Conclusions

From historical data related to past SoS-oriented defense acquisition programs (discussed in Section 2), we summarize the common causes of failure as a) misalignment of objectives among the systems, b) limited span of control of the SoS engineer on the component systems of the SoS, c) evolution of the SoS, d) inflexibility of the component system designs, e) emergent behavior revealing hidden dependencies within systems, f) perceived complexity of systems, and g) the challenges in accurately representing them. These sources of complexity were mapped to a section of the SoSE Process Model recently introduced by Sage and Biemer (2007) in order to identify where manifestations of these complexities might arise and how to begin assessment of how they may impact the acquisition process.

This mapping in conjunction with the 16 technical and technical management SE processes identified by the DoD SoS-SE Guide (Systems and Software Engineering, 2006) was used to develop a conceptual model for pre-acquisition and acquisition strategy activities. The acquisition strategy model takes an incremental approach to the evolutionary development of an SoS and allows processes lower in the hierarchy to affect change in the processes above them. Thus, the model exposes the interconnections among levels and uses these to implement evolving requirements and design solutions in the component systems of the SoS.

These mappings and conceptual models are directed toward providing a basis for a computational exploratory model for acquisition strategy in an SoS environment. Based on user-defined inputs for the requirements and their dependencies on each other, the model uses series and parallel processing to implement these requirements in the component systems. This exploratory model allows evolving requirements and design solutions to trickle through the lower processes and uses disruptors to affect specific component systems, which in-turn affects change in processes higher in the hierarchy.

The uniqueness of the models lie in their ability to provide a better understanding of the acquisition process in an SoS environment, along with computational tools for better decision-making for the higher levels of SoS management. We hope the insights gained from this research will significantly improve the probability of success with future acquisition programs within and without the DoD.

List of References

Allen, T. (2007, April). *Statement on the converted 123-foot patrol boats and changes to the deepwater acquisition program*. USCG Press Release. Retrieved June 1, 2007, from www.piersystem.com/go/doc/786/154307

Anderson, M, Burton, D, Palmquist, M.S., & Watson, J.M. (1999, May). The deepwater project—A sea of change for the US coast guard. *Naval Engineers Journal*. Retrieved April 1, 2008, from http://www.prosoft.tv/aseaofchange.pdf

Bar-Yam, Y. (2003, April). *Complexity of military conflict: Multiscale complex systems analysis of littoral warfare*. New England Complex Systems Institute. Retrieved March 2008, from http://www.necsi.edu/projects/yaneer/SSG_NECSI_3_Litt.pdf

Caldwell, S.L. (2006, June).*Coast guard: Observations on agency performance, operations and future challenges*. Washington, DC: GAO. Retrieved January 3, 2008, from http://www.gao.gov/new.items/d06448t.pdf

Capaccio, T. (2006, December). *Boeing systems delayed, 11 others killed in proposed army budget*. Retrieved June 1, 2007, from http:// www.bloomberg.com/apps/news?pid=newsarchive&sid=amNoYrTtynxQ

Committee on Systems Integration for Project Constellation. (2004 ,September). *Systems integration for project constellation*. The National Academies. Retrieved March 2008, from http://nap.edu/html/proj-constellation/ltr-rep.pdf

DoD. (2006). *Defense acquisition guidebook*. Defense Acquisition Guidebook website. Retrieved April 4, 2008, from https://akss.dau.mil/dag/TOC_GuideBook.asp?sNode=R2-3-2&Exp=Y.

Fowler, C.A. (1994, August). The defense acquisition system too late for the scalpel; bring out the meataxe! *IEEE Aerospace and Electronic Systems Magazine*, *9*(8), 3-6.

Maier, M. (1998). Architecting principles for system-of-systems. *Systems Engineering*, *1*, 267-284.

Polzer, H., DeLaurentis, D.A., & Fry, D.N. (2007, April). Multiplicity of perspectives, context scope, and context shifting events. In *Proceedings of the 2007 IEEE International Conference on System of Systems Engineering* (pp. 1-6).

Riccioni, E. (2005, March). *Description of our failing defense acquisition system as exemplified by the history, nature and analysis of the USAF F-22 raptor program*. Project on Government Oversight. Retrieved March 4, 2008, from http://www.pogo.org/m/dp/dp-fa22-Riccioni-03082005.pdf

Rouse, W. (2007, June). Complex engineered, organizational and natural systems. *Systems Engineering*, *10*(3), 260-271.

Sage, A., & Biemer, S. (2007). Processes for system family architecting, design, and integration. *IEEE Systems Journal*, *1*(1), 5-16.

Saunders, T., Croom, C., Austin, W., Brock, J., Crawford, N., Endsley, M., et al. (2005, July). *System-of-systems engineering for air force capability development* (SAB-TR-05-04). Washington, DC: USAF Scientific Advisory Board.

Simon, H. (1996). *Sciences of the artificial*. Cambridge, MA: MIT Press.

Spring, B. (2005, October). *Congressional restraint is key to successful defense acquisition reform*. The Heritage Foundation. Retrieved March 26, 2008, from http://www.heritage.org/Research/NationalSecurity/bg1885.cfm

Systems and Software Engineering. (2006). *SoS systems engineering*. Retrieved March 2, 2008, from http://www.acq.osd.mil/sse/ssa/initiat_sos-se.html

United States Coast Guard Acquisition Directorate. (1998). *Coast guard recapitalization fact sheet*. Retrieved January 3, 2008, from http://www.uscg.mil/acquisition/programs/pdf/CG-9recap.pdf

Acknowledgements

The authors are grateful for the sponsorship of this work by the Naval Postgraduate School's Acquisition Research Program. The authors also wish to acknowledge and express their gratitude to Chris Patterson and Yury Pensky for their contribution to the project.

THIS PAGE INTENTIONALLY LEFT BLANK

Panel 9 - Implementing Open Architecture and Software Strategies in the DoD

Wednesday, May 14, 2008	**Panel 9 - Implementing Open Architecture and Software Strategies in the DoD**
3:30 p.m. – 5:00 p.m.	**Chair:** **Chair: Mr. Reuben S. Pitts III**, Head, Warfare Systems, NSWC Dahlgren Division **Papers:** ***SHARE Repository Framework: Component Specification and Ontology*** **Ms. Jean Johnson**, Research Assistant and Mr. **Curt Blais**, Research Associate, Naval Postgraduate School ***A Semantic Based Search Engine for Open Architecture Requirements Documents*** **Dr. Craig H. Martell**, Assistant Professor, Naval Postgraduate School ***Emerging Issues in the Acquisition of Open Source Software within the US Department of Defense*** **Dr. Walt Scacchi**, Senior Research Scientist and **Dr. Thomas Alspaugh**, Assistant Professor, University of California, Irvine

SHARE Repository Framework: Component Specification and Ontology

Presenter: Jean Johnson is a research assistant in the Naval Postgraduate School Systems Engineering Department. She coordinates a program to develop Modeling and Simulation curriculum for DoD Acquisition workforce personnel and performs research for the PEO IWS SHARE program. Previously, Johnson held various positions supporting NAVSEA's Warfare Systems Engineering Division. She is a US Navy active and reserve veteran and continues her Navy affiliation in the Individual Ready Reserve. Johnson holds a ME in Operations Research and Systems Analysis and a BS in Applied Mathematics from Old Dominion University and is currently pursuing her PhD in Software Engineering at NPS.

Jean Johnson
Naval Postgraduate School
1 University Circle
Monterey, California 93943
E-mail: jmjohnso@nps.edu
831-656-2956

Author: Curtis Blais is a research associate in the Naval Postgraduate School Modeling, Virtual Environments, and Simulation (MOVES) Institute. His research interests include application of Web-based technologies to improve interoperability of C2 systems and M&S systems. He is contributing to metadata design efforts for the SHARE program, the Department of Defense (DoD) M&S Community of Interest Discovery Metadata Specification, M&S Catalog, and standardized DoD Verification, Validation, and Accreditation (VV&A) documentation, as well as international standardization efforts for the Military Scenario Definition Language and the Coalition Battle Management Language. Blais hold BS and MS degrees in Mathematics from the University of Notre Dame and is a PhD candidate in the MOVES program.

Curtis Blais
Naval Postgraduate School
1 University Circle
Monterey, California 93943
E-mail: clblais@nps.edu
831-656-3215

Abstract

Data sharing is the information technology watchword of our time. Revolutions in information exchange and interoperability are underway in government and industry through policies on the strategic end to data standards on the implementation end. The revolution is transforming acquisition systems and processes through specification of open architectures, which enables construction of new complex systems from crafted components. In August 2006, Program Executive Officer, Integrated Warfare Systems (PEO IWS), established the Software, Hardware Asset Reuse Enterprise (SHARE) repository to enable the reuse of combat system software and related assets. The Naval Postgraduate School (NPS) is tasked to develop a component specification and ontology for the SHARE repository framework. This paper summarizes the work completed to date in support of the PEO IWS 7-sponsored NPS research project "Software Component Specification Framework and Ontology for the SHARE Repository." We describe SHARE and the vision for the repository framework. We present related technologies and the next steps for the framework development. Finally, we provide ideas for future research.

Introduction

Typical challenges for reuse repositories include the lack of motivation for reusable component developers and those wishing to reuse the components, the difficulty of component retrieval and selection, and the amount of work necessary to integrate a selected component. Often, a developer seeking reusable assets from a repository has difficulty finding the desired items due to the inability of the repository designers to foresee what will constitute valuable criteria to guide the search. As a result, the metadata in the repository search tool does not provide adequate information to enable efficient decisions about which assets to pursue. Additionally, most common searches are conducted based on key words and phrases; this requires that the searcher knows how to express desired component features in the terminology employed by the submitter of the asset or the repository manager and its developers.

To further complicate the matter, the goals of reuse depend on the activity being performed within the software lifecycle. To enable reuse during each activity, the desired assets will differ as well as the information required about those assets. For example, during the requirements activities, the sought-after reusable assets may be requirements for similar systems or design artifacts; while at implementation time, developers searching a database may be seeking bits of code or interface specification information. In order for a single repository to support these different searches, the repository builders must incorporate sufficient metadata information to support each type of search. With this perspective, it is clear that the provision of inclusive metadata for components in a database is difficult at best.

In August 2006, the Program Executive Officer, Integrated Warfare Systems (PEO IWS) established the Software, Hardware Asset Reuse Enterprise (SHARE), a library of combat system software and related assets for use by eligible contractors (both prime contractors and subcontractors) for developing or suggesting improvements to Navy Surface Warfare Systems. SHARE is one piece of the Navy’s Open Architecture (OA) approach to developing modular, open systems (PEO IWS, 2007), which includes reusable software applications as a core principle. PEO IWS is currently seeking ways to improve and mature the capability provided by SHARE. Among other initiatives, two related research projects are in progress at the Naval Postgraduate School (NPS). The first, and the topic of this paper, will produce a component specification and ontology for use in SHARE. The second will develop a prototype of a semantically based requirements search engine (ReSEARCH) with the tools necessary to convert documents into semantically based formal representations of requirements (Martel, 2007).

The component specification will describe the artifacts contained in the repository in sufficient detail to aid a repository user in determining whether the artifact is worth retrieving. The ontology will provide contextual semantics describing relationships among items in the repository to aid in associating artifacts with user needs. The component specification and ontology will comprise a rich structural and semantic framework for SHARE that will enable multiple kinds of search and discovery techniques. The goal is to enable the development of different types of tools to improve the usefulness of SHARE.

In this paper, we present the work completed to date on the SHARE ontology and component specification project. We describe the SHARE repository, present our conceptual vision for the repository framework, and describe the technologies that will be used in its development. We also summarize related work and our plans for completion of the framework.

SHARE Repository

SHARE provides a capability for discovering, accessing, sharing, managing, and sustaining reusable assets for the Navy Surface Domain's programs (Belcher, 2007)[1]. It consists of an asset library and a card catalog. The asset library is a collection of combat systems software and supporting artifacts. As of January 2008, 62 assets containing 18,017 artifacts from the Aegis, Ship Self Defense System (SSDS), Littoral Combat Ship (LCS), DDG-1000, and Single Integrated Air Picture (SIAP) programs were available in the SHARE library. The card catalog is a Web-based interface that facilitates user insight into the contents of SHARE and supports user functions—including account registry, asset search and discovery, asset submission assistance, and asset retrieval requests.

The SHARE asset library is separate from the card catalog for two primary reasons. First, the majority of the contents of SHARE is classified material and, therefore, must be kept in a SECRET or higher container. Second, the process for retrieving assets from SHARE includes necessary steps for addressing the data rights associated with each component. For most of the components, a license agreement and Non-Disclosure Agreement are required before an asset can be issued (these forms are also available on the website). Due to these restrictions, the Web interface and the actual assets are physically separated.

The search and discovery process in SHARE is conducted either through individual navigation of the list of assets in the catalog or by a keyword search of the metadata contained in the catalog. From the catalog list, a user can select an asset to obtain a more detailed description, consisting of identity, description and usage information if available.

Assets are requested from SHARE using an online interactive questionnaire. The tool then prepares the necessary documents (including non-disclosure and license agreements) and provides them, along with instructions for printing and submission, to the user. Once the documents have been mailed to the SHARE administrators, the user can track the status of the request online through the SHARE interface. When approved for distribution, library materials are provided either online through access to the appropriate portion of the SHARE website (classified or unclassified) or via delivery of physical media.

A more complete description of SHARE is available in Johnson (2007).

Conceptual Vision for the Software Repository Framework

Popular software repositories, such as SourceForge (2007) and CPAN (2007), tend to be organized to support keyword searches over broad categories of software types. Essentially two types of search are enabled. Items in the repository are grouped by type or function and are then browsable within those categories. A keyword search over metadata

[1] Organizations interested in registering for access to the library should visit and complete an online registration form at https://viewnet.nswc.navy.mil.

is also possible. Repository entries vary greatly in the amount of information (metadata) available for each registered artifact.

The Goal: Improved Search Capability

Current repositories do not support all the types of search we would like to enable. In addition to typical types of search, we envision a graphical user interface that enables navigation of repository assets depending on user's interests. This requires an interface that allows users to project their context on the search mechanisms. In other words, users bring particular information needs and goals based on the problem they are trying to solve. The interface needs to have natural mechanisms to enable users to pose inquiries that fit readily with their views of the problem space. For example, users may be seeking particular functionality best obtained through a functional view or by sorting the information in the repository. Or, users may be seeking particular artifacts best obtained through a document resource organization of the information. Or, users may be seeking information on certain testing methodologies that have been applied so that organization of the information by work activity would best apply. The challenge in designing the framework for the software repository is devising initial sets of such taxonomic descriptions of the assets while creating flexibility for future introduction of additional and diverse organizational views (profiles or templates) of the information as user needs and repository utility grow.

Fish-eye Graph

One example of the type of tool that could be supported by the framework is a fish-eye graph (Sarkar & Brown, 1993). This is a visualization tool that has not, to date, been used to aid in navigation of repository contents. Fish-eye graphs, depicted in Figure 1. display to the user objects of interest in addition to the relationships that the objects have with other items. As the relationships of interest to the user are explored, the graph highlights the item and brings it to the front of the display. The user can then weed out uninteresting items by removing from view the relationships that are not important. This type of search results in a single or small grouping of items that the user has found interesting, with supporting information available by mouse-click.

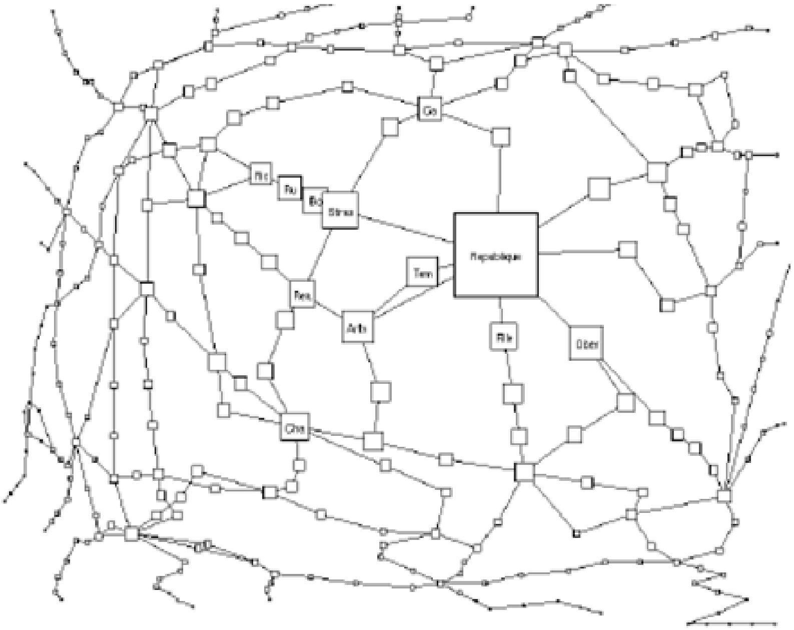


Figure 1. Example Fish-eye Graph
(Sarkar & Brown, 1993)

Semantic Search

Current repository metadata schemas do not address issues of language ambiguity. Rather, they assume that the keywords provided by the metadata will match identically to the words inserted by the user. By providing a framework of related concepts in which to place the artifacts, a search tool can be designed to navigate for artifacts in such a way that users do not need to know the initial set of exact words used to describe the artifacts.

A related ongoing NPS research project titled ReSEARCH is focused on solving these types of issues for SHARE. This work intends to enhance current search mechanisms, principally Latent Semantic Indexing (LSI), by employing word-sense relationships provided in the extensive WordNet lexical database (Princeton, 2006). However, this body of work lacks the domain-specific lexicon found in focused endeavors, such as Navy combat systems. Formalized semantic descriptions in the SHARE component specification and ontology will further enhance ReSEARCH capabilities to produce highly relevant search findings for users of the SHARE repository.

Model-based Search

A third type of search we have envisioned is based on a user-constructed model of the problem the user is trying to solve. The user interface for the repository can provide the capability to assist the user in building the model of a desired system architecture using a standardized representation scheme (e.g., Unified Modeling Language), and the search can then return possible existing solutions for portions of the system and demonstrate potential gaps. Model-based search has similarities to the semantic search concept described above—taxonomic and ontological descriptions of systems, system components, lifecycle phases, development artifacts, usage, and other concepts prominent in the software-hardware domain of SHARE provide structural information that can greatly facilitate search of available assets.

The metadata collected in current repositories do not support these types of advanced discovery tools. Here we provide an overview of our approach for developing a repository that will enable these types of search capabilities.

The Solution: The SHARE Framework

To enable the types of tools we envision, we must create a richer semantic framework for the repository. The framework will be composed of two parts: the component specification and the ontology.

Component Specification

The component specification is a description or model of the items in the repository. For our efforts, we will focus on two aspects of the component specification: “typical” metadata and software behavior.

Metadata—The metadata for each artifact should incorporate all necessary data for discovery and implementation. The metadata will both aid repository users in determining if the item is suited for their use, and provide information about how to use the retrieved asset. We refer to this as “standard” or “typical” metadata since there are many existing examples of metadata that we can use to develop such metadata for SHARE.

Software Behavior—The metadata for many current repositories, such as those described earlier, fail to capture a searchable representation of the functionality of the items outside general categories of functionality (e.g., Archiving Compression Conversion, Control Flow Utilities, Graphics, Security) and text-based search. Unlike current practice, the SHARE component specification will consist of both typical metadata and a behavioral model of the component. Since this piece of the component specification is not commonly incorporated into repositories in a standardized manner, we feel it is a specific focus area to identify the appropriate representation mechanisms for software behavior in the repository context.

Ontology

The second part of the framework is description of the relationships of the components. These form a contextual model of the repository items to represent a particular perspective that can more closely match a user’s problem context. These relationships may include the component’s use/role in existing systems, its mapping to reference or domain architectures, its utility in various software development lifecycle phases, and other types of relationships we expect to discover during the research. Consider the example relationships among artifacts shown in Figure 2. Suppose we insert a requirements document for a particular component into the repository. This artifact may have been originally developed for System A in the figure. The item’s relation to the rest of the original system provides the context for one dimension of the repository framework. If this item was then reused to fulfill some requirements of System B, its location in that model provides a second dimension. Additionally, the requirements document will map to some taxonomy of artifacts that are relevant for particular phases of the product lifecycle. Finally, the component it describes may also have a place in some domain-specific reference architecture. All these relationships provide contextual information about the artifact that can be exploited to enable sophisticated search and discovery methods described above. For this project, an appropriate representation of component context will be identified and the relationships defined. This will enable navigation of the repository based on the contextual information provided in the ontology.

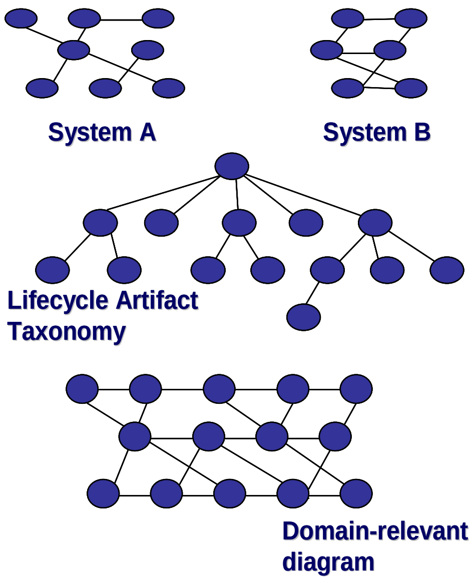


Figure 2. Artifact Relationships

Based on this vision then, the project team has identified three focus areas for developing the framework for the SHARE repository:

1. "Typical" metadata for artifacts
2. A suitable representation of software behavior
3. Framework relationships (ontology)

The current research project will focus on building each of these items for the SHARE repository. Follow-on work will be required to implement the framework in a tool suite that will enable the search capabilities described above. This and other suggested follow-on work is described in the future work section of this report.

Related Technologies

There have been concerted efforts in recent years to add semantics to stored information in order to promote a greater ability to support automated search and sharing of information. The following subsections highlight a number of efforts that can inform our design of the SHARE repository framework.

Metadata Initiatives

There has been much work done on specification of metadata to describe assets and resources in various repositories. For the SHARE framework, we do not expect to create any unique approaches to developing metadata, nor will we develop any fundamentally different metadata set. However, we intend to use the metadata descriptions to support navigation-by-context search, in addition to performing more traditional types of searches based on keywords, text-analysis, and popularity.

World Wide Web

The World Wide Web has experienced unprecedented growth over the past 20 years. This growth was largely fueled by the use of Hypertext Markup Language (HTML), Hypertext Transfer Protocol (HTTP), and Uniform Resource Identifiers (URIs) as simplistic mechanisms for putting information into document files, posting and accessing those files, and linking across those files, respectively. However, HTML primarily describes how the information should be displayed in browser software, rather than providing clear descriptions of the information content of the document. To address this shortcoming, the World Wide Web Consortium (W3C) created the Extensible Markup Language (XML) as a standard way to create and apply markup to the content of Web documents to make the content more readily accessible by software. While initial application of XML made description of Web content much more precise, it largely described content in a structured, syntactic manner. As the demand for greater automation in accessing and processing Web content continued to rise, principal designers and researchers on the Web created a new vision called the Semantic Web.

The Semantic Web is Tim Berners-Lee's vision of the World Wide Web (Berners-Lee, Hendler & Lassila, 2001) in which vast stores of information become meaningful to computers and where "the explicit representation of the semantics underlying data, programs, pages, and other Web resources will enable a knowledge-based Web that provides a qualitatively new level of service" (Daconta, Obrst & Smith, 2003, p. xxi). The Semantic Web is an extension of the World Wide Web in which information is given semantically rich descriptions that enable automated processing by software. The W3C has created additional layers of markup, building on the base of XML, to provide description of the semantics of the information. The Semantic Web is an evolution of the current Web, built from the foundation of open standards on which the Web is built. Building blocks of the Semantic Web are shown in Figure 3. Below, we provide a brief description of the base layers of the Semantic Web stack (URI/IRI and XML) and highlight their relevance to the SHARE metadata development. Applicability of other components of the Semantic Web stack to the SHARE framework will be discussed later in this paper.

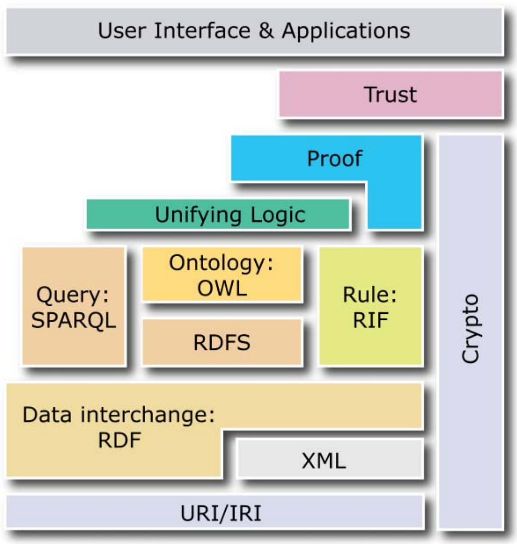


Figure 3. Principal building blocks of the Semantic Web Stack
(W3C, 1994)

Data Sharing Policies in the US Government

In the US Department of Defense, including DoD intelligence agencies and functions, the guiding document for information sharing is the Net-Centric Data Sharing Strategy (DoD Chief Information Officer, 2003). The document defines net-centricity as "the realization of a networked environment, including infrastructure, systems, processes, and people that enables a completely different approach to warfighting and business operations" (DoD Chief Information Officer, 2003, p. 1). The network foundation is the Global Information Grid, "the globally interconnected, end-to-end set of information capabilities, associated processes, and personnel for collecting, processing, storing, disseminating, and managing information on demand to warfighters, defense policymakers, and support personnel" (DoD Chief Information Officer, 2003, p. 1). Data assets addressed by the strategy include system files, databases, documents, official electronic records, images, audio files, websites, and data access services. Users and applications can search for and "pull" data as needed, or they can receive alerts when their subscribed data is updated or changed (publish/subscribe). The goals of the strategy are to make data

- visible—users and applications can discovery the data assets
- accessible—users and applications can obtain the data assets
- institutionalized—data approaches are incorporated into DoD process and practices
- understandable—users and applications can comprehend the data, both structurally and semantically to address specific needs
- trusted—users and applications can determine the authority of the source of the data assets
- interoperable—metadata is available to allow mediation or translation of data to support many-to-many exchanges of data

- responsive to user needs—mechanisms for improvement through continual feedback are supported to address particular perspectives of data users. (DoD Chief Information Officer, 2003, p. 10)

The design of the repository framework should provide or support mechanisms that address each of these goals. In this respect, the data sharing goals help to guide the design and development efforts. A guiding document is the DoD Discovery Metadata Specification (DDMS) (Deputy Assistant Secretary of Defense, 2007), which provides a standard set of metadata for discovering distributed resources. The SHARE repository, as with other repository efforts, can readily address the DDMS by ensuring that sufficient metadata are provided in descriptions of assets to allow generation of at least the minimum required set of metadata specified in the DDMS.

Existing Repository Metadata

Outside the DoD, there are additional metadata practices from which we can learn. All existing repositories have some sort of metadata schema, whether well defined or not. Also, there are some specific efforts that focus on the development of metadata standards for use in software repositories. Some examples of each of these are discussed here.

As introduced earlier, open source repositories such as SourceForge and CPAN have a metadata set for describing their contained assets. Unfortunately, these schemas are not often published. However, they can be somewhat derived simply by looking at the available information for each of the items in the repository.

Another approach is the Object Management Group (OMG) Reusable Asset Specification (RAS) standard for the packaging of software assets. The RAS describes required and optional classes, as well as required and optional attributes, for packaging software assets. The specification is depicted as Universal Modeling Language (UML) models, which are translated into XML Schema and Meta-Object Facility (MOF) / Extensible Metadata Interchange (XMI) XML Schema. In the RAS, artifacts are defined as “any work products from the software development lifecycle,” and assets are a grouping of artifacts that “provide a solution to a problem for a given context” (Object Management Group, 2005, p. 7). Accordingly, the RAS describes an approach for packaging artifacts into an asset using a manifest file.

For SHARE metadata development, we will use these existing examples of metadata as references when the metadata schema is developed. Existing metadata sets will be used to trigger the evaluation of items that could be included but were not originally considered. The goal is not to merge all existing sets of metadata but to assess the relevance of existing data sets for SHARE and include any appropriate items.

Software Behavior Representation

Repositories today tend to capture software behavior as key words describing a general functional area or as a free text description field in the metadata. This type of description can be helpful to users in determining whether the item will be useful in meeting their needs. However, if the desired end-goal is more sophisticated than today’s repository capabilities, a more formal description of behavior is required. For example, one of the loftier goals of a software repository may be to automatically compose systems from reusable components. This is a difficult problem, which many have tried to solve[2]. It is especially difficult if the components were not originally designed for reuse. As a necessary first step towards more sophisticated uses of a repository, behavioral descriptions must be machine-readable in order to support automated search and discovery. Furthermore, the behavior descriptions must be formalized and consistently applied to each item in the repository if the intent is to automatically compose them into a larger functioning system.

A formalized description of software behavior typically means one of two things. We either (1) define the inputs and outputs (interfaces) of the components, or we (2) describe the operations that take place within the component. Many people view the latter as a decomposition of the former. In other words, they describe the inner workings of a component by defining the inputs and outputs of a more granular subset of components. Therefore, here we summarize the current approaches for documenting both software interfaces as well as behavior.

Interface Descriptions

Interface descriptions focus on the inputs and outputs of a component and not the inner workings of that component. Interfaces are represented using various methods, which vary from specific concentration on the connect points between two pieces of software and the types of information passed across them, to representations of the services that a component provides.

One often-employed method for representing interfaces in component software technologies is as a contract between the client and the provider of the implementation (Szyperski, 2002, p. 53). The contract defines the services promised by the interface and the requirements of the client for using the interface. It could simply consist of a set of named operations that can be invoked by clients. It may also include pre- and post-conditions necessary for the successful use of the interface. A drawback for using this type of interface description as a basis for search and discovery in SHARE is the dependency on the component's originating software language for determining the syntax and semantics used to describe the operations and conditions. In SHARE's heterogeneous environment, these types of standardized descriptions may not be practical.

Component technology developers have developed Interface Definition Languages (IDLs) to specify interfaces independently of the programming language used for source code development (Clements et al., 2002, p. 554). Examples include OMG IDL and Microsoft's COM IDL. The same drawback discussed for the programming language-dependent contracts for our heterogeneous SHARE environment exists for these intermediate languages. Rather than a dependency on the programming language, however, the dependence here lies in the chosen component technology. Since we do not intend to force a specific component technology for all SHARE contributors, it does not make sense to insist on interface definitions based on these IDLs.

[2] The proceedings from the International Symposium on Software Composition, an annual event, provide examples of research into the breadth of research topics currently being pursued in the area of software composition. The website for the 2008 conference is located at http://www.2008.software-composition.org/.

Architecture Description Languages (ADLs) are primarily used to formally represent system architectures for use during development and typically describe system elements, their interactions, and their composition rules. While there are many different viewpoints about what constitutes an ADL (Medvidovic & Taylor, 2000, pp. 71-72), they always include a formal description of interfaces. ADL interface descriptions typically define the required and provided services (messages, operations, and variables) of a component. Some ADLs also allow for parameterization of interfaces; others provide additional information.

The advantage of using ADLs in a component specification is that the benefits of ADL-based tools may be realized for the components. ADL tools assist the developer by supporting architecture creation, visualization, validation, refinement, simulation, and analysis, in addition to features that enable systematic transformation of architectures into the implementation of a system. Many support generation of “glue code” for components once their implementations are developed. Additionally, ADLs are likely the appropriate level of abstraction for a heterogeneous collection of assets, such as those found in SHARE, since they do not depend on any decisions made about the implementation of the components.

Unfortunately, the use of ADL-type descriptions does come with a cost. Because of the robust descriptive capabilities of many ADLs, there is considerable effort required in learning how to use them. This would present a learning curve for both asset submitters and retrievers. To minimize this problem, tools could be developed to aid the user in producing the required ADL descriptions. As an alternate solution, we are investigating the possibility of incorporating some ADL-like descriptions into the XML-defined metadata for the components. This will enable us to incorporate only those aspects that are relevant to the SHARE repository. Several new XML-based ADLs such as XML-based Architecture Description Language (XADL) (Zhang, Ding & Li, 2001, pp. 561-566) and Service Oriented Architecture Description Language (SOADL) (Jia, Ying, Zhang, Cao & Xie, 2007, pp. 96-103) may form the basis for this development.

Interfaces can also be represented using UML or other graphical notations. Typical graphical notations of interfaces include the “lollipop” depiction or the expression of an interface as a UML stereotype. Often, these pictorial depictions of interfaces are further defined using a formalized language such as the OMG IDL described earlier (Clements et al., 2002, p. 241). In addition to the visual aid provided by the diagrams, the value of using UML for interface descriptions is that many tools have been developed to read UML and translate the models into XML depictions (XMI) and into executable code. Model-driven Architecture products are available that enable the automatic development of “glue code” between components from the architecture specification (Frankel, 2003).

On the downside, an object-oriented programming development paradigm is assumed. While some generality can by achieved by using packages and subsystems as the main UML building blocks instead of classes and subclasses, some argue that attempting to use UML outside the arena for which it was designed is more trouble than it is worth (Shaw & Clements, 2006, p. 34). This realization, as well as our understanding that whichever description method is chosen must be applied across multiple development cultures, compels us to assert that UML may not be the best way to represent interfaces for SHARE.

Future deployment of the SHARE repository is likely to evolve toward the Service-Oriented Architecture (SOA) of the GIG. SOA has been described as “an ideal vision of a world in which resources are cleanly partitioned and consistently represented" (Erl, 2005, p. 3) and "automation logic is decomposed into smaller distinct units of logic […] known as services" (pp. 23-33). Elements of a service architecture are similar to SHARE concerns—the architecture typically includes a registry of services containing descriptions of services and access information. Mechanisms are provided for service discovery, passing sufficient information about the service back to the caller so that the service can be employed. Advanced concepts include service orchestration for composing higher-order services from component services. The focus, of course, is service reuse, which potentially reduces development and maintenance while improving software reliability and evolution agility.

SOA realization may employ several Web Services standards: Universal Description, Discovery, and Integration (UDDI) for creating service registries; Web Services Description Language (WSDL) for identifying operations offered by services and describing input/output interfaces for those operations; the Simple Object Access Protocol (SOAP) for accessing services and passing data to/from the services; Web Services Business Process Execution Language (WS-BPEL) for describing workflow logic for orchestration of services; OWL for Services (OWL-S), for an ontology of services supporting service advertisement and discovery, description of service operation, service interoperation; Web Services Interoperability (WS-I) profiles for describing collections of Web services specifications at specific version levels; and others. It is interesting to note that the problem of describing Web services in sufficient semantic detail to enable automatic composition of services is similar to the problem of describing software components for reuse.

In Web Service implementations, XML is generally used to hold the information passed across an interface. XML schemas are extensible and easily modified if there is a need to change the standardized format of the data. The above standards for describing and implementing Web Services are XML-based specifications. Additionally, XML is readily digestible by many existing tools and is well enough understood universally to be implemented into new ones. These advantages motivate us to propose XML as the primary notation for documenting metadata, including the interfaces, for the SHARE component specification and ontology project. The flexibility of XML will enable us to incorporate the necessary information to enable capabilities similar to those enabled by ADLs without the high overhead cost of training the end users. Although SOA and Web Services are in a high state of flux as industry standards mature, they present opportunity to create software component specifications in SHARE that can be employed for a number of purposes.

Modeling Software Behavior

In addition to understanding the interfaces for a component, a repository user is interested in the functionality of the software components. In this section, we discuss a number of notations currently used to describe the activities that take place within a component.

In addition to the structural diagramming capabilities provided by the UML, several types of diagrams are used to model dynamic aspects of the system. Methods for formal documentation of behavior provided by UML include sequence diagrams, which may be further amplified using a constraint language such as UML's Object Constraint Language (OCL), collaboration diagrams, and statecharts. Sequence diagrams, or message sequence charts, show the interactions of objects within a component in a time-ordered sequence (Larman, 2005, pp. 222-225). Collaboration, or communication, diagrams also show objects and their interactions but in a more condensed format that tends to lose the visibility of the time-ordered sequencing. State Machine Diagrams, or statecharts, illustrate events and states of objects (Larman, 2005, pp. 485-492). Amplifying information, such as actions triggered by transitions, and activities that take place during particular state conditions is often included.

Each of these UML diagrams sheds light on particular aspects of a component's behavior and could be used to formalize the behavioral descriptions of artifacts incorporated into SHARE. There are a few drawbacks to this approach, however. First, as discussed previously, the use of UML diagrams often assumes an object-oriented development paradigm, which may not be relevant for all SHARE submitters. Second, the UML tools presented are primarily to assist in system development and may not be best suited for asset discovery and retrieval. Repository users are likely to be more interested in a more abstract view of the system than this implementation level information provides. Finally, each of the diagrams only captures a particular "slice," or view, of the software's behavior. For a complete behavioral description, it would be necessary to require each type of diagram plus additional information. This would result in a steep overhead to develop this information for each item contained in SHARE. For these reasons, we do not anticipate incorporating UML activity/state diagrams as the standard representation method for software behavior. However, if these depictions are generated as part of the software engineering development process, they should be included as artifacts in the repository.

In formal specification, system behavior is described using mathematical structures. Formal notations that enable this type of specification include the Vienna Development Method (VDM), Z (pronounced zed), and Alloy. Since the languages are mathematically based, developers can use logic to reason about a formally specified system and sometimes prove its correctness. Application of such techniques generally requires a solid understanding of set theory, logic and other mathematical foundations when learning how to construct specifications in the language. This is one of the complaints about formal languages as well as one of the reasons that the use of formal specification is mostly a topic of research and limited in practical applications to systems or portions of systems with safety-critical reliability demands.

MIT's Alloy project is one of the more successful attempts at making formal methods more user-friendly (Jackson, 2006). Alloy helps users develop the specification by providing a visual simulation of the model. This enables users to recognize when the model is incorrect, and they can then iteratively develop the model in more detail. Alloy also includes an analyzer that automatically checks invariants for inconsistencies in the model. Even with these advances, however, the amount of effort required to specify systems in these formal notations is well above the desired level of effort threshold for the SHARE repository. Therefore, we do not intend to use formal languages to represent software behavior of assets in SHARE.

For SHARE, we do not hope to solve the software composition problem in the near term. Mandating formal descriptions of software behavior for repository items does not seem worthwhile when the composition problem remains unsolved. However, intermediate steps towards formalized behavior descriptions will prove useful in the near term and helpful in advancing towards far-term goals. To this end, we are currently planning to extend the XML-defined metadata to incorporate interface information as well as existing reference architecture information to standardize behavioral descriptions for each artifact entered into the repository. Ongoing advances in service composition in SOAs will also be examined for application to the framework.

Relationships Framework (Ontology)

Rich ontologies capturing the relationships of entities from multiple views have not been applied to software repositories. However, there are many examples of ontology use in the organization of data for different applications. One example is intelligence community synthesis of disparate pieces of information from widespread sources into logical connections in order to form coherent pieces of knowledge. There are currently several applications designed to collect the data and assist the analyst in drawing relationships among the data. For example, Palantir Technologies has created one such software application to support the DoD intelligence community by providing robust capabilities for managing data from various sources. The Palantir tool is based on user-defined ontologies and supports multiple representation and analysis tools. Graphical representations depict the data items and their relationships with each other, based on the underlying ontology. The analysis tools can be used to form logical links between entities in the database and to detect patterns and irregularities in the data. This rich environment enables multiple search techniques: using keywords, browsing through data tables, and browsing graphical views of the database content based on the relationships of the entities described in the ontology.

For the SHARE research project, these capabilities serve as examples of potential utility of the repository framework by demonstrating the power of formalized semantics. When the framework is in place, technologies such as these can be exploited to gain flexibility in the search options described previously. Similar examples of the use of ontologies to support data analysis exist in other domains, particularly in the medical field. Some background on current and emerging standards for describing rich semantics in data relevant to the SHARE framework is provided in this section.

Semantic Web Techniques

The Semantic Web stack was shown in Figure 3. Several of the components pictured there contribute to stronger semantic description of Web-based resources and are discussed briefly here.

The Resource Description Framework (RDF) is a language for stating assertions in the form of subject-predicate-object triplets. Each of the elements in an RDF statement is an abstract Web resource identified by a URI. RDF and RDF Schema (RDFS) will be investigated for applicability to the SHARE framework to describe taxonomies (class hierarchies) supporting inference and search (Alesso & Smith, 2006). We will also explore the possible benefits of creating RDF expressions for storing SHARE repository data content.

The lower layers of the Semantic Web stack provide the ability to describe information (metadata and schemas) and to express knowledge (assertions). Query languages provide a means to access information. The XML Query language is used to search XML documents by exploiting the hierarchical tree structure of the documents (XPath expressions). The SPARQL Protocol and RDF Query Language provide a means to search RDF expressions by exploiting the subject-predicate-object graph structure of the expressions (pattern matching). If RDF structures prove valuable for describing information in the SHARE repository, the use of SPARQL and other query techniques will be explored.

The Web Ontology Language (OWL) extends RDF/RDFS constructs to provide more precise description of classes, subclasses, and relationships among classes (properties). OWL adds the capability to define local scope of properties, disjointness of classes, Boolean combinations of classes, cardinality restrictions, special characteristics of properties (e.g., functional, transitive, symmetric), and other aspects not expressible with RDF/RDFS (Alesso & Smith, 2006). We will investigate the use of OWL for ontology development for the SHARE framework. Use of OWL will maximize utility by software applications, including use of openly available reasoning engines that can be used to check for ontology consistency and to make inferences about instances in the asset knowledge base.

Rules and rule-based systems provide additional expressiveness in describing the logic of a system. Rules permit software to infer a conclusion from a premise (Alesso & Smith, 2006). Rules may be used in the formalized specification of software assets in the repository to enrich their description, particularly if there is a need to encode business rules, policies, and processes appropriate to the repository (e.g., role-based access). The use of the well-established, Web-based conventions in the information technology community provides a basis for application of a variety of common logical computations. We will be able to employ existing products that can operate on the semantic descriptions using provably correct methods. Cryptologic aspects of the Semantic Web stack cut across all the layers and support such functionality as authentication, encryption, and digital signature (Eastlake & Niles, 2003). We will not address this area directly in the work, but will be creating the semantic basis for implementation of methods such as role-based access and other controls on information content in the repository. Trust is obtained when we can anticipate the actions of a system and have a reasonable expectation that the system will act correctly (i.e., as intended) (Michael, 2008). Trust is often established and maintained through transparency. One of the advantages of the use of the Semantic Web practices is visibility of the information through its description in metadata, semantic descriptions, rules, and computationally sound logic. Clearly, users of the repository will rely on the trustworthiness of the content when obtaining information or artifacts supporting new developments. While we will not address this aspect of the problem directly in the component specification and ontology development, our goal is to make the information as explicit and accessible as possible to humans and machines in order to promote this level of the Semantic Web stack.

Well-defined syntax and semantics for description of metadata, taxonomies, and ontology for the SHARE framework will facilitate development of software applications and user interfaces for working with the repository. By expressing the SHARE component specification and ontology using common Semantic Web elements, the products of our current research will readily support development of various applications including Web Services in an SOA while also providing a basis for future applications employing emerging Semantic Web Services technologies.

Semantic Search

Semantic search methods “augment and improve traditional search results by using not just words, but meaningful concepts” (Alesso & Smith, 2006, p. 201). A prominent approach is Latent Semantic Indexing, which considers documents that share many words in common to be semantically close, without any understanding of the “meaning” of the words. As introduced earlier in this report, other researchers at NPS are developing semantic search capabilities (ReSEARCH) for the SHARE repository that will use the WordNet database to extend this approach to include related words (synonyms, part-of relationships, etc.). For even greater formulation of context, the metadata, taxonomy, and ontology specifications for the SHARE framework discussed above will provide domain-specific semantics that should enable more precise discernment of relevance in the searches. As the formalized semantics of the component specification and ontology are developed, the formalisms will be provided to the ReSEARCH developers to determine if improvements in search precision can be achieved.

Enriched semantic specification of the assets in the SHARE repository will enable users to more readily find resources that meet their need in their context. Extensive work in the Web community is providing tools and techniques that can be applied to the SHARE framework. We will select and apply appropriate techniques to meet the goals of the framework development.

Next Steps

Based on our vision for the framework and the related existing technologies we have summarized, this section lays out our intended path for completing development of the SHARE repository framework.

SHARE Metadata

An initial list of required asset information has been developed by the SHARE Program Office at Naval Surface Warfare Center, Dahlgren, VA. We have developed an XML schema based on this initial list and will complement the metadata fields with the necessary information for filling out the framework. To fill out the data set, we will evaluate known good metadata examples and pull relevant information into the SHARE metadata. We will then ensure that the metadata includes all the information necessary to place the artifact in the appropriate context based on the ontology. In order to promote maximum exposure of SHARE contents, we will also ensure that minimum requirements of the DDMS are satisfied. Based on these considerations, we will develop a practical metadata schema. This will most likely include a core data set and variations for different types of artifacts.

In order to evaluate the completeness of the metadata, we intend to investigate case studies for each phase of the software development cycle. As stated previously, repository users' needs vary greatly depending on their purpose at the time of search. Therefore, we are constructing case studies that capture the potential needs based on users' current development activities. For each of these case studies, the metadata will be evaluated to ensure inclusion of appropriate information for enabling retrieval decisions.

SHARE Software Behavior Representation

For the SHARE software behavior representation, we suspect that the overall goal of implementing formalized representations of standardized software behavior is not feasible in the short term. While we intend to keep the loftier goal in mind, it is likely that an interim step towards standardization of formal software behavior representation will be required.

One near-term solution may be to use available domain information that standardizes descriptions of software functionality. For example, the Common Systems Function List (CSFL), Common Operational Activities List (COAL), and Common Information Element List (CIEL) are leadership-endorsed listings of combat system functionality that can be utilized as an initial characterization of software behavior. We will investigate the use of a subset of these listings in the development of taxonomies for the SHARE repository framework. If we require asset submitters to state the functionality of the components in these terms, we can then build the tools to guide the user in selecting desired behavior in the same terms. We will also explore characterization of software assets based on current and emerging Web Services (e.g., WSDL) and Semantic Web Services (e.g., WS-BPEL, OWL-S) approaches.

SHARE Relationship Framework (Ontology)

The ontology for SHARE will be based on several relationships among the items in the repository, as well as relevant domain architectural descriptions and other information. The types of relationships we are currently exploring are the artifact's place in the software engineering lifecycle, its architectural fit in its original system, its architectural fit in any systems in which it was subsequently used, identification of the component's fit in the Surface Navy Open Architecture reference architecture, and the semantic relationships of various documents in the repository (based on the ReSEARCH work). Each type of relationship will be examined to determine its appropriate representation form (RDF, OWL, etc.). The goal is to determine representation forms that will best enable tool development, which will in turn support the types of search described in the previous chapters based on the ontology provided.

Future Work

The current research will describe the component specification and ontology desired for the SHARE repository. Further work will be necessary to implement the framework and develop a tool suite that enables the described search capabilities. In the SHARE implementation, additional repository features can be added, such as an Amazon-like "similar results" feature that points people with similar problems to the retrieval of the same files (and other similar recommendations found in Johnson (2007)). In the long term, further work will be required if the intent is to eventually enable automated composition of a system based on reusable components. As mentioned previously, a starting point to accomplish this goal may be to standardize a formal behavior representation of the repository contents.

List of References

Alesso, H.P., & Smith, C.F. (2006). *Thinking on the web: Berners-Lee, Gödel, and Turing.* Hoboken, NJ: John Wiley & Sons.

Belcher, M. (2007). *PEO IWS software hardware asset reuse enterprise (SHARE)*. Information brief.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001, May 17). The semantic web. *Scientific American*, *284*(5), 34-43.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., et al. (2003). *Documenting software architectures: Views and beyond*. Boston: Addison-Wesley.

Comprehensive PERL Archive Network (CPAN). (2007). CPAN. Retrieved January 16, 2007, from http://www.cpan.org

Daconta, M.C., Obrst, L.J., & Smith, K.T. (2003). *The semantic web: A guide to the future of XML, web services, and knowledge management*. Indianapolis: Wiley Publishing.

DoD Chief Information Officer. (2003, May 9). *Net-centric data sharing strategy*. Washington, DC: Author.

Deputy Assistant Secretary of Defense. (2007, August 10). *Department of Defense discovery metadata specification (DDMS)* (Ver. 1.4.1). Washington, DC: Deputy Chief Information Officer.

Eastlake, D.E,. III, & Niles, K. (2003). *Secure XML: The new syntax for signatures and encryption*. Boston: Addison-Wesley.

Erl, T. (2005). *Service-oriented architecture: Concepts, technology, and design*. Upper Saddle River, NJ: Pearson Education.

Frankel, D.S. (2003). *Model driven architecture: Applying MDA to ENTERPRISE COMPUTING*. Indianapolis: Wiley Publishing.

Jackson, D. (2006). *Software Abstractions*. Boston: MIT Press.

Jia, X., Ying, S., Zhang, T., Cao, H., & Xie, D. (2007). A new architecture description language for service-oriented architecture. In *Proceedings from the 6th International Conference on Grid and Cooperative Computing* (pp. 96-103). Washington, DC: IEEE Computer Society.

Johnson, J. (2007, October). *SHARE repository component specification: Needs assessment.* Technical Report. Monterey, CA: Naval Postgraduate School.

Larman, C. (2005). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

Martel, C. (2007). *ReSEARCH: A requirements search engine*. Proposal for Future Combat Systems Open Architecture for Naval Sea Command PEO IWS-7.

Medvidovic, N., & Taylor, R.. (2000). A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering*, *26*(1), 70-93.

Michael, B. (2008, March 19). Perspectives, practices, and the future of building highly dependable and trustworthy systems. Software Engineering Presentation delivered at the Naval Postgraduate School, Monterey, CA.

Object Management Group. (2005). Reusable Asset Specification (Ver. 2.2). Retrieved January 29, 2008, from http://www.omg.org/technology/documents/formal/ras.htm

Sarkar, M., & Brown, H. (1993). Graphical fisheye views. *Communications of the ACM*, *37*, 73-83.

Shaw, M., & Clements, P. (2006). The golden age of software architecture. *IEEE Software*, *23*(2), 31-39.

SourceForge. (2007). *SourceForge.net: Welcome to SourceForge.net.* Retrieved October 8, 2007, from www.sourceforge.net

Szyperski, C. (2002). *Component software: Beyond object-oriented programming* (2nd ed.). New York: Addison-Wesley.

Princeton University. (2006) *WordNet—Princeton University Cognitive Science Laboratory*. Retrieved March 20, 2008, from wordnet.princeton.edu

World Wide Web Consortium (W3C). (1994). *Semantic web stack.* Retrieved February 26, 2008, from www.w3.org/2007/03/layerCake.png

Zhang, B., Ding, K., & Li, J. (2001). An XML-message based architecture description language and architectural mismatch checking. In *Proceedings from the 25th Annual International Computer Software and Applications Conference* (pp. 561-566). Washington, DC: IEEE Computer Society.

Acknowledgements

The work described in this paper is sponsored by PEO IWS. The authors would also like to thank Dr. Mikhail Auguston for his contribution to the preparation of this paper by providing leadership in developing the concept for the repository framework. The opinions expressed in this paper are those of the authors and not necessarily the position of their respective parent organizations or the sponsor of the work described.

ReSEARCH: A Requirements Search Engine

Presenter: Craig Martell holds a PhD from the University of Pennsylvania and is an Associate Professor in Computer Science at the Naval Postgraduate School. He specializes in natural-language processing applied to on-line chat, weblogs and semantic search.

Dr. Craig Martell
Associate Professor
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
E-mail: cmartell@nps.edu

Author: Paige H. Adams, Lieutenant, is a US Navy Information Warfare Officer. He has served as a Russian and Burmese linguist aboard a variety of surface, subsurface, and airborne platforms in the Atlantic, Pacific, and Arabian Gulf, as well as at joint field sites and on flag staff. Most recently, he served as Operations Officer and Joint Operations Department Head at the Misawa Security Operations Center in Japan. He is a 2000 graduate (summa cum laude) of Hawaii Pacific University with a BS/BA in Computer Information Systems.

LT Paige Adams, USN
Graduate Student
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
E-mail: phadams@nps.edu

Author: Dr. Pranav Anand is an Assistant Professor of Linguistics at UC Santa Cruz. He specializes in theoretical syntax and semantics, with an emphasis on how contextual factors affect the form and meaning of linguistic expressions. He received his AB from Harvard in 2001 and his PhD from MIT in 2006.

Dr. Pranav Anand
Assistant Research Professor
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
E-mail: panand@nps.edu

Author: Grant Gehrke, Ensign, is a native of Louisville, Kentucky. He graduated in 2007 from the Naval Academy with a degree in Computer Science. His current research focuses on using NLP approaches for authorship attribution in blogs. He will be graduating from the Naval Postgraduate School in June and reporting to Forrest Sherman, a destroyer homeported in Norfolk, Virginia.

ENS Grant Gehrke, USN
Graduate Student
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
E-mail: gtgehrke@nps.edu

Author: Dr. Ralucca Gera received her BS (2000), MA (2002), and PhD (2005) in Mathematics, all from Western Michigan University. She has been an Assistant Professor in the Applied Mathematics Department at NPS since 2005. Gera's research interests are in combinatorics and graph theory, particularly domination and alliances in graphs.

Dr. Ralucca Gera
Assistant Professor
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
E-mail: rgera@nps.edu

Author: Marco Draeger is a Captain in the German Army and studied Computer Science at the University of the Federal Armed Forces from 2000 to 2004. From 2004 to 2005, he was assigned to the Modeling and Simulation Department of the Center for Transformation of the Federal Armed Forces. Following this, he served at the German Command Support School until 2007. Since June 2007, Draeger has been a student in the Operations Research Department at the Naval Postgraduate School.

Marco Draeger
Graduate Student
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
E-mail: mdraeger@nps.edu

Author: Dr. Kevin Squire received his PhD degree from the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (UIUC), Urbana, in 2004. He is currently an Assistant Professor of Computer Science at the Naval Postgraduate School, Monterey, CA. His research focuses on developing and applying machine-learning algorithms and models to many different domains—including natural language processing, autonomous language acquisition, scheduling, robotics, computer vision, and remote sensing.

Dr. Kevin Squire
Assistant Professor
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943
E-mail: kmsquire@nps.edu

Abstract

This research address three closely related problems. (1) Most current search technology is based on a popularity metric (e.g., PageRank or ExpertRank), but not on the semantic content of the document. (2) When building components in a service-oriented architecture (SOA), developers must investigate whether components that meet certain requirements already exist. (3) There is no easy way for writers of requirements documents to formally specify the meaning and domain of their requirements. Our goal in the research presented here is to address these concerns by designing a search-engine that searches over the "meanings" of requirements documents. In this paper, we present the current state of the ReSEARCH project.

Keywords: Semantic Search, Requirements, Open Architecture, Information Systems Technology

1 Motivation

While modern computing has made it possible to access enormous amounts of information with little effort, much of that information comes without any indexing, making manual search of it all but impossible. The science of *information retrieval* (IR) attempts to correct for this by extracting information from a collection of documents based upon a search request, or *query*. While the field of IR has focused a great deal of attention on how the form, or *syntax*, of a query and the documents in the collection can aid the process of extracting information, it has paid far less attention to the meanings, or *semantics*, of those forms.

Semantic analysis can be computationally intensive, and for certain domains, sensitivity to meaning may not provide a system with sufficient improvement to justify the greater computational cost incurred. However, there are at least two conditions in which semantically sensitive search can lead to improvements over keyword-based approaches: a) when the document collection is composed of human-generated free-text, and b) when the document collection is in a specialized domain, with non-standard terminology and assumptions (where the standard for most IR is the general content of the World Wide Web).

The Software, Hardware Asset Reuse Enterprise (SHARE) repository card catalog is in the intersection of both conditions mentioned above. The SHARE card catalog should ideally allow a user to search for an asset based upon free-text overviews generated during asset submission, as well as additional structured metadata (Johnson & Blais, 2008). Because this overview is written in free-text, the syntactic form in which the information expressed by the overview cannot be guaranteed in advance, making search over it quite difficult. In addition, the elements being searched over are descriptions of military assets. So, the document collection for this IR task is in a specialized domain, and the search process should be sensitive to the semantic connections that are particular to this domain.

In order to appreciate the challenges posed by IR over free-text and in specialized domains, we now turn to the complications that each condition brings to the task.

1.1 Challenges of Free-Text Search

Human language in general has several properties that make information retrieval taxing. Formally speaking, any language, human or man-made, can be expressed as a relation between *form* (syntax) and *meaning* (semantics); thus fluency in a domain consists of knowing the relation between the forms of the language and their corresponding meanings. Man-made languages often aim to make this relation as straightforward as possible. For instance, in the mathematical language of arithmetic the syntactic symbol "+" stands for the semantic concept of numerical addition. However, note that the symbol "−" can stand for two different semantic concepts: numerical subtraction or the marking of negative numbers. Thus, “−” has multiple meanings, and we say that it is *polysemous*. In arithmetic, only “−” is polysemous, but in human languages polysemy is pervasive. The word *tank*, for example, has multiple meanings (or, *senses*)—it may refer to weaponry or a water tank. In the information-retrieval context, polysemy renders a query (and sentences in the document set) ambiguous: if the user is searching for tank specifications, are they asking about water tanks or weaponry?

Polysemy complicates the form-meaning relation by having multiple possible meanings for a given word. In addition, human language routinely has multiple words attached to a given meaning. We call these *synonyms*. For example, the verb *consume* has many synonyms, e.g., *devour, ingest, eat*. If a user enters the query “What type of fuel does an F-22 consume?”, without an understanding of synonymy, the system will not be able to return to the user, for example, a document containing the answer “The F-22A Raptor uses JP-8”. Hence, synonymy complicates the information-retrieval task by creating the (quite likely) possibility that the meaning requested by the user is expressed in a different form than the one the user used in the query. Synonymy may occur at all levels of linguistic form; for example, the sentences, “The F-22A uses JP-8” and “JP-8 is the fuel type for the F-22A Raptor” convey the same semantic information despite their rather different forms. In particular, the first sentence lacks a synonym for *fuel*, meaning that sentence-level synonymy cannot simply be the product of word-level synonymy.

One final complication of searching over human language is that the relationships between semantic entities are not necessarily represented in the syntactic forms of the entities. For instance, the semantic entities *mother* and *daughter* are connected by a parental relation. In order to determine the sentential synonymy of *Mary is Jane's mother* and *Jane is Mary's daughter*, the system must understand the the relationship between *mother* and *daughter*. This is a rather challenging task if we are simply looking at linguistic form, as there is nothing in the words *mother* and *daughter* that indicates they are connected. Such information is accessible only once we have some representation of the meanings of the words (or larger elements), and some way of deriving inferences between them.

1.2 Challenges of Domain-dependent Search

As detailed in (Johnson & Blais, 2008), the SHARE repository asset library currently consists of combat systems software and supporting artifacts, but will become more diverse (e.g., through the incorporation of hardware components). The card catalog will thus contain information about the specification and function of such artifacts. As Johnson and Blais note, there is (and will continue to be) a high level of similarity between the SHARE artifacts, given that they are all specifiable under the Surface Navy OA Warfare Systems Architecture Element Level Decomposition. Hence, their overviews will share many characteristics atypical of documents found on the Web, making Web-based tools sub-optimal.

However, this speciality of SHARE's domain could prove an advantage for semantically sensitive search. For instance, it could allow for a reasonably robust polysemy control. For example, in the SHARE context, a query involving *consume* is more likely to refer to fuel usage than to eating. Similarly, the domain could aid semantic inferencing (of the sort exemplified by the pair *mother-daughter*) based both on terms in the free-text overview and the larger functional context of a particular asset. Hence, based on facts regarding the objects under discussion within SHARE, the system could conclude that there is a relation between *ballistics* and *shell-size*, allowing searches regarding one to consider documents containing the latter. Additionally, building on the product lines that assets play in the Navy enterprise, the system could infer that a given asset possesses certain properties that may be useful the user.

1.3 Domain-independent Learning for Domain-dependent Rules

Given that the polysemy and inferencing subsystems we are building are particular to the specialized domain of SHARE, one natural question is how such subsystems will be developed? One possibility is to build a handwritten set of rules, and have the IR system look to those rules when performing inferencing, such as that implemented in the Wordnet project.[1] While such strategies are undoubtedly useful, they typically: a) are time-consuming, b) lack empirical coverage (human error may cause a rule to go unnoted), and c) require constant supervision for a dynamic document collection. All three pitfalls are of concern with regard to SHARE; the most troubling is probably the requirement for constant maintenance, given that SHARE is an evolving repository and a potential model for similarly constrained repositories over different kinds of assets.

Given such problems, we propose that the domain-dependent components of ReSEARCH be generated not by human input but by machine learning over the document collection of SHARE and, in the initial stages, additional informational resources. The goal of ReSEARCH is to develop a system of tools for determining domain-dependent resources to address the issues surrounding polysemy, synonymy and inference.

The remainder of this paper will detail the problems of contemporary approaches to IR and our investigations of approaches to integrate semantically sensitive tools. In the following section, we will present an overview of common approaches to IR, and why they will fail in dealing with collections such as SHARE. We will then discuss the algorithmic issues involved in generating tools that allow semantically sensitive searching. Finally, we will present the current status of our implementation of the ReSEARCH system.

[1] Wordnet is an ongoing project directed by George Miller at Princeton University's Cognitive Science Laboratory to encode relations between semantic entities. It may be accessed at http://wordnet.princeton.edu.

2 Prior and Current Strategies

The Web is a tremendously useful repository of information. Unfortunately, this information is unstructured, and there is no canonical "Table of Contents" or "Index," making web search one of the most challenging of today's Internet problems. Two attempts were made to address this challenge: (1) hand-classified directories (as originally used by Yahoo, for example), and (2) query-based search engines (for example, AltaVista and, eventually, Google). This second class is what concerns us here. For more details on Web Search Engines see (Schwartz, 1998).

Search engines employ a centralized architecture in which so-called "spiders" collect website information, and an indexer makes an index of these pages to ease the search. In the early 1990s, the first phase of web search was simply keyword search. In keyword-based search, all pages containing requested keywords are returned, ranked according to the strength of match (e.g., the number of times a word appears in a document, if it is in the title, etc.).

AltaVista used this strategy originally. In 1995, it was the first company to fully index the visible pages on the World Wide Web. Over time, it evolved different search modes: basic search, advanced search, and power search (Notess, n.d.). One advanced feature that AltaVista and other search engines added was stemming (Sapp, 2000). Stemming ensures that words with plurals and suffixes (e.g., -ed, -ing, -er) are always treated as being in their stem form (Hersh, 2003, p. 178). Unfortunately, it is unclear how useful stemming is in the search process (Harman, 1991).

The second phase in Web search was the development of techniques that used the connection between pages to create a ranking of the websites for more accurate search. The indexing problem was changed into finding the most appropriate way to "rank" each website. One easy solution was to make the rank proportional only to the number of other pages linking to the page in question. However, this ranking method turned out to be inaccurate for a variety of reasons. In particular, it did not take into account the source of the links, allowing someone to easily boost the rank of a page by increasing the number of incoming links, thus subverting the indexing mechanism (Langville & Meyer, 2006).

To avoid this index subversion, new methods needed to be developed which took advantage of the link structure of both the Web and the meaning of the queried word so the output was most relevant to the query. The challenge, then, was to increase the relevance of the returned pages to the query itself.

2.1 Page Rank and Expert Rank

In 1998, Google revolutionized search. They did this not by changing the fundamentals, as the pages returned are still those that match the keywords in the query, but by changing the order in which the return pages were presented. Google ranked all pages according to the a then-novel ranking algorithm called PageRank (Brin & Page, 1998).

The essence of the Google innovation is in how the PageRank algorithm works. The rank of each page in a search depends not only on the number of pages pointing to it, but also on the rank and the number of outgoing links of these pages. To further determine the rank of all web pages, Google simulates the behavior of virtual surfers randomly surfing the web. A page's rank is then updated based on how frequently the random surfers visit that page. This pre-existing rank of each individual website is assigned independently of any query. As a result of this ranking, the pages are ranked in order of sociological importance: the more links with higher weight there are to a page, the more important it is in the "society" of pages. Additionally, hubs—pages that have a lot of links pointing to them—are given greater authority. In other words, the importance of a link is determined by both the rank of the linking page and the number of outgoing links from that page. One pitfall of this scheme (which Google attempts to corrects) is that communities of websites can trap random surfers, which in turn, increases the rank of those websites.

As an example of how this is implemented, let "Q" be a query, a list of words, that is saved as a vector $\vec{q}$, whose binary components show whether a particular word is present or not in Q. Also, the information on each website $R_1, R_2, \ldots$ is similarly saved as the binary vectors $\vec{r}_1, \vec{r}_2, \ldots$, respectively. By computing the inner product $\langle \vec{q}, \vec{r}_i \rangle$, $i \geq 1$, or the cosine similarity measure using the normalized vectors corresponding to the vectors above, the system can identify the similarity between the query and the pages in the universe. However, this method does not take into account the correlation between websites and their semantics. To overcome this problem, PageRank uses the *PageRank metric* $PR(P)$ that defines recursively the rank/importance of each page P by

$$PR(P) = (1 - d) + d(\sum_{i=1}^{n} PR(T_i)/C(T_n)),$$

where d is a damping factor (0.15, as used by Brin and Page in (Brin & Page, 1998)), T_i ($1 \leq i \leq n$) are all the pages pointing to P, and each T_i has $C(T_i)$ outgoing links. So, P receives a fraction of the weight $PR(T_i)$, as this weight is equally spread among all the outgoing links from T_i, for $1 \leq i \leq n$ (Zhang & Dong, 2000).

Ask.com, formerly known as "Ask Jeeves," is another search site offering state-of-the-art search, this time based on technology called "ExpertRank" (Ask.com, n.d.). In addition to examining the number of links entering a site, ExpertRank also attempts to identify topic clusters related to a search, as well as experts within these topics, and use all of this information to rank search results.

2.2 The State of Online Search Using Natural Language Processing

Since the "Semantic Web" has become a buzzword in the Internet community and in business at large, several organizations have emerged to provide "Semantic Search." Many promising companies and research projects have built search systems that crawl the web for annotated data over which to search, such as web sites with RDF data. This search strategy, however, does not allow searching documents that do not have rich, hand-built, metadata. In particular, the vast majority of documents online, written in natural human language, are not searched. A small subset of these search engines, however, have begun tackling the problem of searching documents consisting only of written language, extracting semantic meaning.

Powerset Labs (www.Powerset.com), a San Francisco-based startup, has positioned itself as a forerunner in this field by attempting to leverage natural language processing in their search system. Currently honing their search algorithm, Powerset indexes and searches Wikipedia for question-answering tasks. The documents in this database are written in plain text and, for the purposes of search, do not contain extended metadata. Instead, the Powerset indexing algorithm identifies linguistic features such as *named entities* and *parts of speech* to improve search results.

Being a private, for-profit company, the Powerset search algorithm is not public, but some important functionality can be extracted from public demonstrations. The Powerset labs website currently contains two methods of searching Wikipedia. The first is a general search of the document index, which encourages queries to be phrased as questions. Queries such as "When did earthquakes hit San Francisco?" and "politicians from Virginia" are among the suggested queries. Results of these queries return results that demonstrate term matching on a higher level than keyword search. For example, the Powerset system uses "When" as a wildcard to match dates and times that appear in phrases describing earthquakes in San Francisco. "From" is used, in the second example, to search for phrases that indicate some named entity is "from" Virginia. This improves results significantly over a search with just the keywords "politicians" and "Virginia," as are used in standard search engines.

The search "politicians from Virginia" also reveals that "politicians" matches terms such as "governor" and "senator", indicating that an ontology is used to match the term "politician" with its hyponym, "governor." The search "What do zombies eat?" reveals that the Powerset algorithm also searches over synonyms by returning results containing the synonymous verb "devour." This system does not perform rich disambiguation, however, as evidenced by the result "…zombie finishes college," in which "finishes" is considered a synonym of "eat".

Finally, results from the Powerset search "What do zombies eat" include phrases in which the information about what zombies eat is encoded in more complex sentence structures. Correct results such as "granddaughter eaten by zombies," "zombies …where they are brought back from the dead by supernatural or scientific means, eat the flesh or brains of the living", and "His corpse is thrown over the fence to be devoured by the zombies", all reveal that powerful parsing of the sentences is performed in the indexing process rather than strictly requiring matching phrases such as "zombies eat *". Though their indexing structure is not known, the "PowerMouse" demonstration allows the user to search the fact index more directly, confirming that these relationships exist in the indexing for fast searching, eliminating the need for computationally expensive parsing with every search query.

Powerset is thus building capabilities for semantically sensitive search similar to those of ReSEARCH. However, it is not clear that Powerset's approach is designed to handle the domain-specificity of collections like SHARE, meaning that it is not clear their technology can be leveraged to construct novel inferencing mechanisms in particular domains.

3 Automated Inference-rule Discovery

Recall that natural languages, unlike formal taxonomic structures, contain inherent ambiguity of both form and meaning. It is this ambiguity that presents a challenge for natural language applications such as information retrieval or question answering. Two questions arise: 1) which meaning of a word or phrase in a search term does the requester intend, and 2) how do we return results that are related to the search query, even if the search term does not contain the exact word or words? The first question is related to the problem of *word sense disambiguation* and is, itself, a well-studied area. We shall turn our attention to the second problem: *inference*.

3.1 Semantic Similarity from Distributional Similarity

In (Hearst, 1992), Hearst explored using one kind of inference rules to generate others, given a body of text. Specifically, she considered how use of synonymy relations could be used to learn the relation of *hypernymy*, or subtype classification. For concreteness, consider the pair *vehicle-Humvee*. As *Humvee* is a subtype of *vehicle*, the latter is a hypernym of the former. How could a machine learn the hypernymy relation of *vehicle-Humvee* automatically? Hearst's approach exploited the fact that the co-occurrence of words in patterns of the type X *such as* Y, as well as its synonyms X*, including* Y, and Y *and other* X, implies a hypernymic relationship between X and Y. As she demonstrated, if a system were seeded with various synonyms for forms that demonstrate hypernymy, the system could induce hypernymic connections from the text provided.

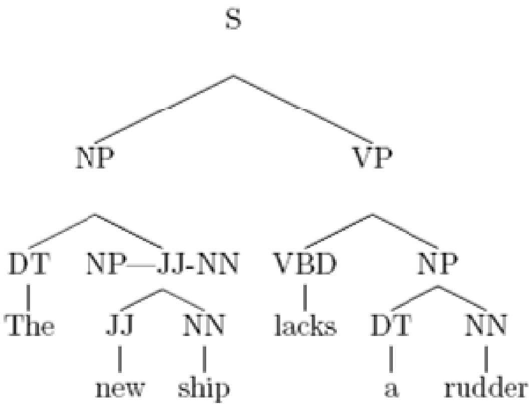


Figure 1. Parse Tree from NLTK Demo

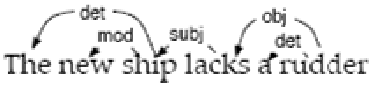


Figure 2. Dependency Tree of Same Sentence as in Figure 1

While Hearst's method is useful for learning various inferences, it relies upon human-generated synonyms for expression of hypernymy (or the relation in question). More desirable would be a system that learns the synonyms themselves from the text, especially given the possibility that such synonyms could be domain-dependent. In their 2001 study, "DIRT—Discovery of Inference Rules from Text"(Lin & Pantel, 2001), Lin and Pantel outline an unsupervised method of discovering inference rules from text, based on the idea that semantic similarity is generally correlated with syntactic similarity. We turn to this next.

3.2 Dependency Trees and Paths

A *dependency* relationship is an asymmetric binary relationship between two words: a **head** and a **modifier**. One can observe the structure of a sentence by examining the tree formed of the dependency relationships contained therein. The tree structure arises from the characteristic that a given word may have more than one modifier, but each word may modify a maximum of one word. Note that a dependency tree differs from a parse tree, which is concerned with the *syntactic* relationship between words. A comparison of the two are shown in Figures 1 and 2.

Dependency graphs are constructed by using Lin's MINIPAR, a broad coverage English language dependency parser (Lin, 2008). Links in the graph represent indirect *semantic* relationships between two words. A dependency path is constructed by joining the words and their link dependency relationships, excluding the two end words. For instance, in our example sentence the dependency path between the words *ship* and *rudder* would be represented by the path N:subj:V←lacks→:V:obj:N. The words *ship* and *rudder* fill the slots in the path at either end. Non-slot dependency relations are called internal relations. In this manner, one can construct the paths of all word pairs in a given corpus of text.

Lin and Pantel (Lin & Pantel, 2001) imposed a set of constraints on the paths to be extracted:

- The "slot fillers" must be nouns, since these are variables that will be instantiated by entities.
- Dependency relations that do not connect the two content words (e.g., in the case of determiners or modifiers), will be excluded from the path.
- There will be a lower limit (threshold) on the frequency count of an internal relation.

To accumulate the frequency counts of paths in a corpus, a **triple database** was used. A triple is comprised of $(p, Slot, word)$ for two words w_1 and w_2. Correspondingly, each such pair of words has two corresponding triples: $(p, SlotX, w_1)$ and $(p, SlotY, w_2)$. $SlotX, SlotY$ and w_1, w_2 are *features* of path p.

3.3 Path Similarity

As alluded to above, Lin and Pantel's approach makes an assumption based on Harris's *Distributional Hypothesis* (Harris, 1954), which assumes that two words will have a similar meaning if they appear in similar contexts. Instead of words, Lin and Pantel assume that the hypothesis also holds for paths between words; i.e., if multiple dependency tree paths link the same set of words, then the meanings of the paths are likely similar. They termed this the *Extended Distributional Hypothesis*.

Computing similarity between two paths first takes into account the **mutual information** between a path slot and its filler. The approach is similar to calculating a $tf \cdot idf$(*term frequency* × *inverse document frequency*) measurement and is performed for a similar reason: to discount high frequency words that may not have the same importance as less frequent words. Pantel and Lin's formula leverages the similarity measurement proposed in (Lin, 1998), but is modified to take paths into account:

$$mi(p, Slot, w) = \log \frac{|p, Slot, w| \times |*, Slot, *|}{|p, Slot, *| \times |*, Slot, w|}.$$

The mutual information thus defined, the similarity between a pair of slots is defined as:

$$sim(slot_1, slot_2) = \frac{\sum_{w \in T(p_1,s) \cap T(p_2,s)} (mi(p_1, s, w) + mi(p_2, s, w))}{\sum_{w \in T(p_1,s)} mi(p_1, s, w) + \sum_{w \in T(p_2,s)} mi(p_2, s, w)}.$$

In this formula, p_1 and p_2 are paths, s is a slot, and $T(p_i, s)$ is the set of all words that fill the s slot of path p_i. Finally, the similarity of two paths p_1 and p_2 is defined by the geometric average of the similarities of their $SlotX$ and $SlotY$ slots:

$$S(p_1, p_2) = \sqrt{sim(SlotX_1, SlotX_2) \times sim(SlotY_1, SlotY_2)}.$$

Comparison of paths in a corpus is accomplished via pairwise comparison of each path using the preceding formulae. Since comparison of all paths is computationally expensive, Lin and Pantel use a filtering algorithm that only compares paths where a candidate path's shared features with an input path p exceed a fixed percentage. This procedure ultimately produces a list of paths in descending order of their similarity to p.

3.4 Results

Lin and Pantel (2001) used MINIPAR to parse approximately 1GB of newspaper text from the *AP Newswire*, *San Jose Mercury-News*, and *The Wall Street Journal*. From this, they extracted seven million paths, 231,000 of them unique, which were then stored in a triple database. For evaluation, they used the first six questions of the TREC-8 Question-Answering Track, extracted the paths from the questions, and generated a Top-40 Most Similar list using their algorithm to determine if the generated paths might contain the answer to the questions posed. This output was also compared to a set of publicly available, manually generated paraphrases of the TREC questions. In the evaluation, a path was deemed to be correct if it was likely that the path could generate the correct response to the question, given that the answer could be found in some corpus. An example used by Lin and Pantel (2001) was the path *"X manufactures Y"* generated from the TREC question, *"What does the Peugeot company manufacture?"* One of the Top-40 most similar paths is *"X's Y factory."* Since *"Peugeot's car factory"* is a likely phrase in some corpus, this generated path is classified as correct.

The DIRT algorithm performance varied widely for different paths. It was noted that paths with verb roots tended to perform better than verbs with noun roots since noun root paths tend to occur less often. Lin and Pantel (2001) also found that, even with high-scoring correct paths, there was little overlap between these automatically generated paths and the manually generated paraphrases, suggesting the difficulty for humans in the paraphrase-generation task. As noted earlier in studies of manual inference-rule generation, completeness errors exist due to the difficulty of paraphrase recall for humans. In this capacity, the DIRT algorithm shows promise in augmenting a manual-generation workflow.

Q#	Paths	Man.	DIRT	Int.	Acc.
Q_1	X is author of Y	7	21	2	52.5%
Q_2	X is monetary value of Y	6	0	0	N/A
Q_3	X manufactures Y	13	37	4	92.5%
Q_4	X spend Y	7	16	2	40.0%
	spend X on Y	8	15	3	37.5%
Q_5	X is managing director of Y	5	14	1	35.0%
Q_6	X asks Y	2	23	0	57.5%
	asks X for Y	2	14	0	35.0%
	X asks for Y	3	21	3	52.5%

Table 1. A Summary of Lin and Pantel's DIRT Algorithm Results on TREC-8 Questions.

A summary of DIRT results on the TREC data is in Table 1. The column labeled "Man." indicated the number of manual paraphrases generated for the question. The next column shows the number of paths found by the DIRT algorithm. The intersection of those two is in the fifth column. The final column shows the evaluated accuracy of the automatically generated paths.

3.5 Related work

Snow et al. (Snow, Jurafsky, & Ng, 2005) leverage a similar method of automated inference-rule discovery using dependency paths in a continuation of the hypernym discovery method pioneered by Hearst. This method involved using the dependency paths in a feature count vector and conducting a binary classification of hypernymy for word pairs based on vector-distance measurement. The results obtained represented a 16% F-score improvement over previous models, and a 40% improvement when augmented with **coordinate terms** (i.e., terms that share a common hypernym ancestor).

4 Implementation issues

Lucene Java is an Open Source project, available under the Apache License, which provides an accessible API for the development of search applications. Lucene provides plenty of opportunities to construct a semantic search engine. A good overview and documentation is available from the Apache Lucene website (*Lucene-java Wiki*, 2008).

A search application developed with Lucene consists of the same two major components mentioned in Section 2: an indexer and a searcher. The indexer builds an index of the given documents; the structure and content of this index depends on the implementation of the indexer application. Typical contents would be the title of a document, its path, a URL, or the actual text content. Content can be stored in different ways, depending on if it has to be searchable or not. The search application typically converts a search string given by the user into a query and then searches the index for matching items. Later in this section, two short examples will demonstrate these processes.

4.1 Interesting Features of the Lucene API

One remarkable property of Lucene is its flexibility. By overriding the stemming and analyzing algorithms, its behavior can be changed into something completely new, particularly from a keyword search engine into a semantic search engine similar to Powerset; however, a very useful property of Lucene is its accessibility from different environments, e.g., Python.

PyLucene is a Python extension for accessing Java Lucene. This extension allows developers to implement some functionality of the desired application using NLTK, a widely used Python-based project for natural language processing. Documentation and implementation samples for PyLucene can be found in Vajda (2005).

Another very helpful feature is a package for indexing and query expansion based on WordNet synonyms. Using the WordNet application, this package creates a synonym index of words and converts search strings into queries which can be used by Lucene. For our first tests, we built an index of synonyms from WordNet and used it to expand and convert search strings into Lucene-compatible queries.

4.2 The Wikipedia Corpus

For our experiments we decided to use downloadable Wikipedia content (`http://download.wikimedia.org/enwiki/20080312/enwiki-20080312-pages-articles.xml.bz2`).

The size of this file is about 60 GB. This size requires an event-based parser such as SAX. For the first experiments only about 160 MB (more than 12,000 articles) from a partial download were used.

The structure of the XML file is as follows: every article is stored in a `<page>` node, which has several child nodes. From these child nodes we used the `<title>` and `<text>` fields. The special syntax of a Wikipedia page was ignored at first, meaning that all the content of an article was given the same priority–particularly we did not distinguish between headings, links or normal text.

Parsing and indexing 12,738 articles took about four minutes on a Windows Vista PC with an AMD64 CPU and 1 GB memory under non-benchmark conditions.

4.3 Sample Implementations

Two sample implementations will be introduced: a Wikipedia indexer and a small search application.

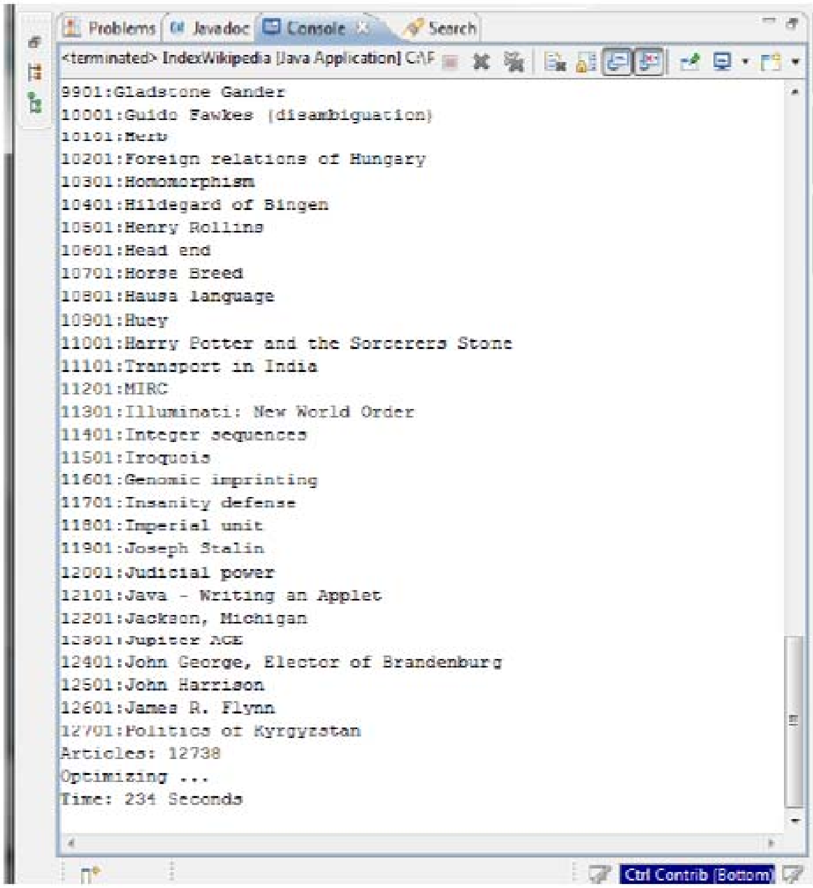
The indexer follows a sample given in (Schmidt, 2005). The original version had to be changed in order to obtain compatibility to the current version of Lucene. Only the main concepts will be considered at this point; for further explanations of the different classes involved, see (*Lucene-java Wiki*, 2008) and Lucene's Javadoc. The main part of an indexing application is the index writer. It writes the index into a file system and also optimizes its structure for faster access. Logically, the written index consists of documents; in our case, every Wikipedia article is treated as a separate document. A document is then split into different fields; for the sample application, these fields were "title" and "text." The indexer determines whether and how a field is stored in the index. The choices are: (1) not to store at all, (2) to store, but not to index, (3) to store and to index it without first analyzing it, and (4) to store it and to index it using an analyzer. An analyzer implements a certain policy for extracting index terms from text. Lucene already implements various analyzers; we used a StandardAnalyzer for our tests. Since an analyzer determines how the content of a document is represented in the index, it provides an opportunity for the developer to implement semantic strategies for building and searching the index. The Wikpedia indexer first parses the xml file, which contains the articles and extracts all `<page>`-nodes, from which the `<title>` and the `<text>` nodes are extracted. Then, for every article, two new fields are generated: "title" and "text." These fields are added to a new document, which is then passed to the IndexWriter-object. After this process, the index content is optimized by the writer, concluding the indexing process.

The searcher was implemented in Python using PyLucene, using a StandardAnalyzer. To be able to search for an article, the user's search string has to be converted into a query. This conversion is done by a Lucene class called QueryParser, which is generated using the name of the field that contains the actual content and the analyzer. The query is then passed to the searcher, which returns an object called "hits." This object holds a list of all matching documents with an assigned score. For our purposes, the searcher application just prints the titles of the matching articles, followed by their score.

The score is assigned by an object which extends the Scorer class in the API. The scorer itself uses a similarity implementation which is based on the cosine-distance between document and query vectors in a Vector Space Model of Information Retrieval.

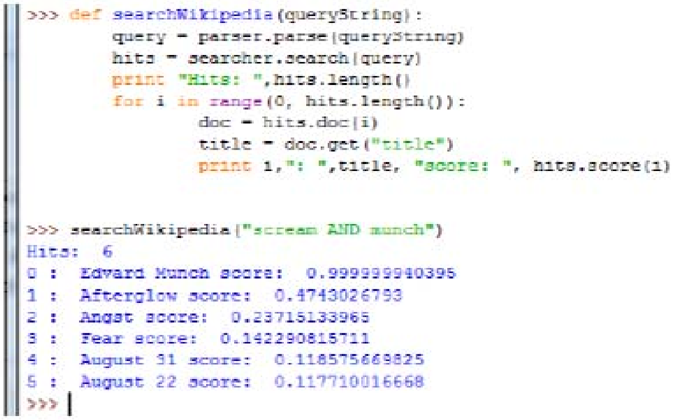
4.4 WordNet Query Expansion Sample

Figure 4.4 shows the output of a standard keyword search using only a single word in the search string versus the output of the same search after expanding the search with the WordNet interface. Only results with a score higher than .5 were printed. This very simple sample



```
9901:Gladstone Gander
10001:Guido Fawkes (disambiguation)
10101:Herb
10201:Foreign relations of Hungary
10301:Homomorphism
10401:Hildegard of Bingen
10501:Henry Rollins
10601:Head end
10701:Horse Breed
10801:Hausa language
10901:Huey
11001:Harry Potter and the Sorcerers Stone
11101:Transport in India
11201:MIRC
11301:Illuminati: New World Order
11401:Integer sequences
11501:Iroquois
11601:Genomic imprinting
11701:Insanity defense
11801:Imperial unit
11901:Joseph Stalin
12001:Judicial power
12101:Java - Writing an Applet
12201:Jackson, Michigan
12301:Jupiter ACE
12401:John George, Elector of Brandenburg
12501:John Harrison
12601:James R. Flynn
12701:Politics of Kyrgyzstan
Articles: 12738
Optimizing ...
Time: 234 Seconds
```

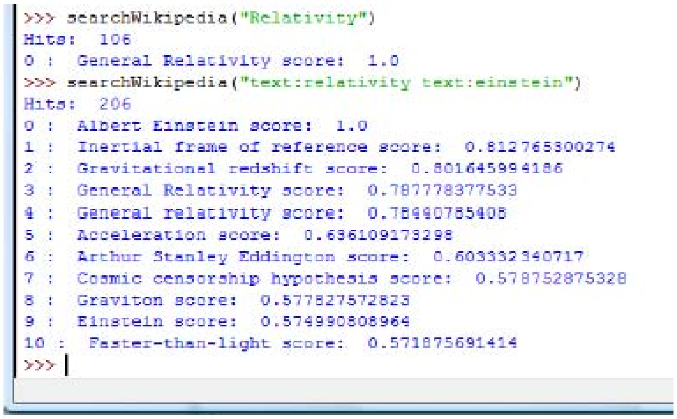
Figure 3. Output of the Wikipedia Indexer



```
>>> def searchWikipedia(queryString):
        query = parser.parse(queryString)
        hits = searcher.search(query)
        print "Hits: ",hits.length()
        for i in range(0, hits.length()):
                doc = hits.doc(i)
                title = doc.get("title")
                print i,": ",title, "score: ", hits.score(i)


>>> searchWikipedia("scream AND munch")
Hits:  6
0 :  Edvard Munch score:  0.999999940395
1 :  Afterglow score:  0.4743026793
2 :  Angst score:  0.23715133965
3 :  Fear score:  0.142290815711
4 :  August 31 score:  0.118575669825
5 :  August 22 score:  0.117710016668
>>> |
```

Figure 4. Searcher Output



```
>>> searchWikipedia("Relativity")
Hits:  106
0 :  General Relativity score:  1.0
>>> searchWikipedia("text:relativity text:einstein")
Hits:  206
0 :  Albert Einstein score:  1.0
1 :  Inertial frame of reference score:  0.812765300274
2 :  Gravitational redshift score:  0.801645994186
3 :  General Relativity score:  0.787778377533
4 :  General relativity score:  0.78440785408
5 :  Acceleration score:  0.636109173298
6 :  Arthur Stanley Eddington score:  0.603332340717
7 :  Cosmic censorship hypothesis score:  0.578752875328
8 :  Graviton score:  0.577827572823
9 :  Einstein score:  0.574990808964
10 :  Faster-than-light score:  0.571875691414
>>> |
```

Figure 5. Comparison of Results Using a Standard Versus an Augmented Search String

shows how using synonyms can improve a search significantly. Note, the Wikipedia article "Relativity" does not appear although it should do so with a score of 1.0. The explanation for this phenomenon is quite simple: the article is not in the corpus–because all experiments were applied on only 12,000 articles, which is less than 1.7% of the actual corpus. To get a perfect hit by a single keyword is, therefore, very unlikely.

4.5 Open Issues

The Lucene API provides several access points through which it can be extended to a semantic search engine. Future work will determine how a document has to be represented in an index to enable a semantic search. This will involve implementation of an analyzer representing the policy for extracting index terms from the corpus. In order to match queries against documents, the analyzer will need to transform search strings into a representation compatible with that of the documents in the index. Additionally, in order to rank documents matching the query according to a scale of relevance, we will need to implement a semantically sensitive scorer.

A final issue of research is how to use WordNet for query expansion beyond addition of synonyms. One relation between words that is worth considering is certainly the hypernym-hyponym relation. WordNet already provides a definition for this relation as a Prolog file. Therefore, a parser for the different WordNet files should be included in the implementation.

5 Conclusion

The ReSEARCH project is still in its beginning stages. However, we have made great strides in identifying the fundamental issues involved in semantic search and how we will need to deal with them in the context of SHARE. Our next step is to start experimentation with proxy data–the Wikipedia data referred to above–and to plan how to move towards *live* SHARE data as it becomes available. Another important aspect of the project that must be handled next is what the *summary* field of the SHARE card catalog must contain.

List of References

Apache Lucene. (2008). *Lucene-java wiki*. Retrieved March 23, 2008, from http://wiki.apache.org/lucene-java

Ask.com. (2008). Ask search technology. Retrieved March 28, 2008, from http://about.ask.com/en/docs/about/ask technology.shtml

Brin, S., & Page, L. (1998, April). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international world wide web conference* (pp. 107-117). Published in *Computer Networks and ISDN Systems*, *30*(1-7).

Harman, D. (1991). How effective is suffixing? *Journal of the American Society for Information Science, 42*(1), 7-15.

Harris, Z.S. (1954). Distributional structure. *Word, 10*, 146-162.

Hearst, M.A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on computational linguistics* (pp. 539-545). Morristown, NJ: Association for Computational Linguistics.

Hersh, W.R. (2003). *Information retrieval: A health and biomedical perspective.* New York: Springer.

Johnson, J., & Blais, C. (2008). Share repository framework: Component specific and ontology. In *Proceedings of the fifth annual acquisition research symposium.* Monterey, CA: Naval Postgraduate School.

Langville, A.N., & Meyer, C.D. (2006, July). *Google's PageRank and beyond: The science of search engine rankings.* Princeton, NJ: Princeton University Press.

Lin, D. (1998). Extracting collocations from text corpora. In *Workshop on computational terminology* (pp. 57-63). Montreal, Canada.

Lin, D. (2008). *MINIPAR*. Retrieved March 28, 2008, from http://www.cs.ualberta.ca/~lindek/minipar.htm

Lin, D., & Pantel, P. (2001). DIRT—Discovery of inference from text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 323-328). New York: ACM.

Notess, G.R. (n.d.). *Review of altavista.* Retrieved March 23, 2008, from http://www.searchengineshowdown.com/features/av/review.html

Sapp, G. (2000). *Altavista proffers updated search technology to online businesses.* Retrieved March 23, 2008, from http://www.infoworld.com/articles/ic/xml/00/03/21/000321icalta.html

Schmidt, M. (2005). *Lucene wikipedia indexer.* Retrieved March 23, 2008, from http://schmidt.devlib.org/software/lucene-wikipedia.html

Schwartz, C. (1998, September). Web search engines. *Journal of the American Society of Information Science, 49*(11): 973-982.

Snow, R., Jurafsky, D., & Ng, A.Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In L.K. Saul, Y. Weiss & L. Bottou (Eds.), *Advances in neural information processing systems 17* (pp. 1297-1304). Cambridge, MA: MIT Press.

Vajda, A. (2005). *Pulling java lucene into python: Pylucene.* Retrieved March 23, 2008, from http://chandlerproject.org/PyLucene/Paper

Zhang, D., & Dong, Y. (2000). An efficient algorithm to rank web resources. *Computer Networks, 33*(1-6), 449-455.

Emerging Issues in the Acquisition of Open Source Software within the US Department of Defense

Presenter: Walt Scacchi is a senior research scientist and research faculty member at the Institute for Software Research, University of California, Irvine. He received a PhD in Information and Computer Science from UC Irvine in 1981. From 1981-1998, he was on the faculty at the University of Southern California. In 1999, he joined the Institute for Software Research at UC Irvine. He has published more than 150 research papers and has directed 45 externally funded research projects. In 2007, he served as General Chair of the 3rd IFIP International Conference on Open Source Systems (OSS2007), Limerick, IE.

Walt Scacchi
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-34 55 USA
Phone: 1-949-824-4130
Fax: 1-949-824-1715
E-mail: wscacchi@ics.uci.edu

Author: Thomas Alspaugh is an Assistant Professor of Informatics in the Donald Bren School of Information and Computer Sciences, University of California, Irvine. He received his PhD in Computer Science from North Carolina State University in 2002. His research interests are in software engineering and focus on informal and narrative models of software at the requirements level. Before completing his PhD, he worked as a software developer, team lead, and manager at several companies (including IBM and Data General) and as a computer scientist at the Naval Research Laboratory on the Software Cost Reduction project, also known as the A-7E project.

Thomas A. Alspaugh
Institute for Software Research
University of California, Irvine
Irvine, CA 92697-34 55 USA
Phone: 1-949-824-4130
Fax: 1-949-824-1715
E-mail: alspaugh@ics.uci.edu

Abstract

In the past five or so years, it has become clear that the US Air Force, Army, and Navy have all committed to a strategy of acquiring software-intensive systems that require or utilize an “open architecture” (OA) and “open technology” (OT) that may incorporate OSS technology or OSS development processes. There are many perceived benefits and anticipated cost savings associated with an OA strategy. However, the challenge for acquisition program managers is how to realize the savings and benefits through requirements that can be brought into system development practice. As such, the central problem we examine in this paper is to identify principles of software architecture and OSS copyright licenses that facilitate or inhibit the success of an OA strategy when OSS and open APIs are required or otherwise employed. By examining and analyzing this problem, we can begin to identify additional requirements that may be needed to fulfill an OA strategy during program acquisition.

Introduction

Interest within the US Department of Defense (DoD) and military services in free and open source software (OSS) first appeared in the past five or so years (Bollinger, 2003). More recently, it has become clear that the US Air Force, Army, and Navy have committed to a strategy of acquiring software-intensive systems across the board that require or utilize an “open architecture” (OA) and “open technology” (OT), which may incorporate OSS technology or OSS development processes (Herz & Scott, 2007). Why?

According to Riechers (2007), the Air Force sees several factors within its software-intensive systems: there is increasing complexity of the software (code) itself; the Air Force may be “held hostage” by proprietary legacy components; it seeks more timely delivery of new solutions, and it is aware that acquisitions and requirements take too long. So, the Air Force is moving towards an OT development approach that embraces open standards, open data, open program interfaces, best-of-breed OSS, and OSS development practices.

According to Brig. Gen. Justice (2007, March; 2007, December), the Army seeks to move away from closed source software, expensive software upgrades, vendor lock-in, and broadly exploited security weaknesses. Subsequently, the Army seeks to adopt OSS because it may realize direct cost savings (compared to proprietary closed source software), gain access to source code in order to better develop domain and IT expertise, enable the transition to Web 2.0 technologies, and enable rapid injection of innovative concepts from diverse R&D/IT communities into systems for tactical command and control (C3T), future combat systems, enterprise information systems, and others (Starett, 2007).

Last, according to Guertin (2007), the Navy seeks to mitigate the spiraling costs of weapon systems through adoption of OA (US Navy, 2006), as well as the adoption of open business models for the acquisition and spiral development of new systems. This may, therefore, necessitate better alignment of the system requirements and program acquisition communities, as well as better alignment of industry and academic partners who engage in software-focused research and development activities with DoD support.

The central problem we examine and explain in this paper is the identification of principles of software architecture and OSS copyright licenses that facilitate or inhibit the success of the OA strategy when OSS and open APIs are required or otherwise employed. This is the knowledge we seek to develop and deliver. Without such knowledge, program acquisition managers and Program Executive Offices are unlikely to acquire software-intensive systems that result in an OA that is clean, robust, transparent and extensible. This may frustrate the ability of program managers or program offices to realize faster, better, and less expensive software system acquisition, development, and post-deployment support.

On a broader scale, this paper seeks to explore and answer the following kinds of research questions: How does the use of OSS components and open APIs (a) facilitate or (b) inhibit the ability to develop and deliver an OA software system? How do the requirements for OA affect system acquisition? How do alternative OSS licenses facilitate or inhibit the development of OA systems? How does the use of OSS components and open APIs manifest requirements that (a) facilitate, or (b) inhibit program acquisition?

Last, this paper may help establish a foundation for how to analyze and evaluate dependencies that might arise when PMs seek to develop software systems that should embody an OA—especially when different types of OSS components or OSS component licenses are being considered for integration. Finally, we believe there are new ways for determining requirements for how best to develop software systems with OSS (Scacchi, 2002) that can interact with acquisition processes (Choi & Scacchi, 2001) in ways that are not apparent within current public perspectives for OA, based on OSS (Guertin, 2007; Justice, 2007, March; 2007, December; Riechers, 2007).

In the remainder of this paper, we examine what makes achieving OA and OT difficult from a technical and program management/acquisition perspective, with respect to understanding what OA incorporating modern OSS entails from a software architecture standpoint, software licensing regimes, and how/where they interact. We start by providing additional background on “openness.” We then add a description and analysis of open software architecture concepts and of open source software licenses. This gives rise to a discussion that identifies new requirements that must be addressed by program managers in acquisitions that are intended to realize an OA software system. We then close with a review of the conclusions that follow.

Background

Across the three military services within the DoD, OA means different things and is seen as the basis for realizing different kinds of outcomes. Thus, it is unclear whether the acquisition of a software system that is required to incorporate an OA as well as utilize OSS technology and development processes for one military service will realize the same kinds of benefits anticipated for OA-based systems by another service (Wheeler, 2007). Somehow, DoD acquisition program managers must make sense of or reconcile such differences in expectations and outcomes from OA strategies across the DoD. Yet, there is little explicit guidance or reliance on systematic empirical studies for how best to develop, deploy, and sustain complex software-intensive military systems in the different OA and OSS presentations and documents that have been disseminated (Weathersby, 2007). Instead, what mostly exists are narratives that serve to provide and promise the potential of OA and OSS without consideration of what socio-technical challenges may lie ahead in realizing OT, OA, and OSS strategies.

In characterizing the challenges facing acquisition of OA and OSS systems, we have found it helpful to compare the new property of “Openness” with the familiar property “Correctness”; we summarize this with the maxim “*open* is the new *correct.*”

Acquisition officers are familiar with the challenges of acquiring systems that meet the necessary requirements with regard to correct behavior. The correctness of the overall system depends on the correctness of its components and how they are interconnected; correctness is a relative quality, in that a system may meet its behavioral requirements to a greater or lesser degree, but almost by definition, a system is never completely correct, and its degree of correctness cannot be definitely established in a finite time. A lack of correctness has an effect when that part of the system is executed (and the correctness of a system in meeting its requirements is determined) by engineers and the system’s users through testing it and using it. Openness is both similar to and different from correctness, however. We argue that the openness of a system depends, like correctness, on the system’s components: how they are interconnected and how they are configured into an overall software system architecture. Unlike correctness, however, a system may be completely open, or may fail to be open in various ways. Because the software elements that define a system are finite and enumerable, its openness can, in principle, be determined. Also unlike correctness, a system is either open or not open even when it is not operating, and DoD may pay the consequences of a lack of openness (in the form of license fees) before the system is ever used or even if it is never used. Finally, unlike correctness, openness may—ultimately—be the province of lawyers and policy makers, not engineers or users.

We believe that a primary challenge is how to determine whether a system, composed of sub-systems and components—each with specific OSS or proprietary licenses and integrated in the system's planned configuration—is or is not open, and what license(s) apply to the configured system as a whole. This challenge comprises not only evaluating an existing system, but also planning for a proposed system to ensure that the result is "open" under the desired definition, with only the acceptable licenses applying. It is also important to understand which licenses are acceptable in this context. Because there are a range of licenses (each of which may affect a system in a different way), and due to the number of various kinds of OSS-related components and ways of combining them (which have an effect on the licensing issue), the first step in this process is to understand types of software elements that constitute a software architecture, and the types of licenses that may encumber these elements or their overall configuration.

OA seems to simply suggest software system architectures incorporating OSS components and open application program interfaces (APIs). But not all software system architectures incorporating OSS components and open APIs will produce OA, since OA depends on: (a) how/why OSS and open APIs are located within the system architecture, (b) how OSS and open APIs are implemented, embedded, or interconnected, (c) whether the copyright (Intellectual Property) licenses assigned to different OSS components encumber all/part of a software system's architecture into which they are integrated, and (d) whether many alternative architectural configurations and APIs may or may not produce an OA (Alspaugh & Antón, 2007; Diallo, Sim, & Alspaugh, 2007; Scacchi, 2007). Subsequently, we believe this can lead to complex situations: if program acquisition stipulates a software-intensive system with an OA and OSS, then the resulting software system may or may not embody an OA. This can occur when the architectural design of a system constrains system requirements—that is, which requirements can be satisfied by a given system architecture when requirements stipulate specific types or instances of OSS (e.g., Web browsers and content management servers) to be employed, or what architecture style (Bass, Clements & Kazman, 2003) is implied by given system requirements.

Thus, given the goal of realizing an OA and open technology strategy (Herz & Scott, 2007), together with the use of OSS components and open APIs, it is unclear how to best align program acquisition, system requirements, software architectures, and OSS license regimes.

Understanding Open Software Architecture Concepts

A system intended to embody an open architecture using open software technologies like OSS and APIs does not clearly indicate which possible mix of software elements may be configured into it. To help explain this, we first identify the types of software elements included in common software architectures, whether they are open or closed (Bass et al., 2003).

- *Software source code components*—These include the computer programs that direct the intended computation, calculation, control flow, and data manipulation.

These are programs for which the source code is open for access, review, modification, and possible redistribution by their developers. However, there are currently at least four forms of computer programs.

- *standalone programs*—These are the computer programs that we have long understood, often as isolated systems or monolithic applications that accept data inputs, manipulate and transform this data, and produce outputs (calculated results, information displays, emit control signals to devices, etc.) under user or system administered control.
- *libraries, frameworks, or middleware*—These are collections of software functions, no one of which is typically a standalone program. Such software is often expected to be routinely reused in many different systems or applications. This software may also be used to provide a layer of abstraction that hides source code implementation details so as to improve subsequent software portability, or to hide alternative software implementations.
- *inter-application script code*—This software is used to combine independent programs by associating their respective inputs, outputs, and control variables. This software is sometimes called “glue code,” which suggests its primary use is to connect programs through the use of “pipes” and/or “filters” that control or modulate the directed flow of information between the associated programs. Such scripts may be as short as a single line of code, but on the other hand, they can be as large as thousands (even hundreds of thousands) of source lines of code.
- *intra-application script code*—This software is similar in spirit to inter-application script code, except the focus is on organizing, controlling, and manipulating input and output data/presentations from remote Web services/repositories for view and end-user interaction at the human-computer interface. Popular Web application systems like the Firefox Web browser may be scripted to provide animated user interfaces coded in languages like Javascript, ActionScript, or PhP to create Rich Internet Applications (Feldt, 2007) or “mashups” (Nelson & Churchill, 2006). Such scripts may be as short as a single line of code, but on the other hand, they can be as large as thousands (even tens of thousands) of source lines of code. However, custom intra-application software languages may also be designed to create domain-specific languages (e.g., XUL for Firefox Web browser (Feldt, 2007)) for rapid construction of persistent/disposable software functions (or macros), which enable increased software development productivity or end-user programming.

- *Executable components*—These are programs for which the software is in binary form, and its source code may not be open for access, review, modification, and possible redistribution. Executable binaries are rarely treated as open since they may also be viewed as “derived works” (Rosen, 2005) that result from the compilation or interpretation of software source code that may not be available, or may be proprietary. Executable components are widespread and common in every computing system, even in OSS systems. However, executable components may also only become part of a system during its execution through dynamic (or run-time) linking. Finally, though their binary form makes them available for execution through external linkage to some other program, such form also makes figuring out what they do very difficult, if they have little/no documentation available.

- *Application program interfaces/APIs*—These software interfaces are generally not programs that can be executed, but they enable software system developers to access their functionality without direct access to their source code. The availability of externally visible and accessible APIs to which independently developed components can be connected is the minimum required to form an “open system” (Meyers & Obendorf, 2001). Often, the APIs are treated as if they enable direct access to the otherwise hidden software, but a closed software system may employ a layer of abstract APIs as “shims” that better align multiple program interfaces or security barriers that seek to protect disclosure of private or proprietary information. Such information may include the details of actual software function interfaces (which may be designated as “trade secrets”) or hidden software functions that may only be known to software developers with secure, restricted code access.

- *Software connectors*—These may be software either from libraries, frameworks, or application script code whose intended purpose is to provide a standard or reusable way of associating programs, data repositories, or remote services through common interfaces. These may include software technologies that constitute a “software bus” for plugging in independent software modules (programs or functions), network protocols that enable and control the flow of data between remote programs across a LAN or Internet, or even a database management system (DBMS) that is used to enable data sharing and storage among programs connected to the DBMS. The High Level Architecture (HLA) is an example of a software connector scheme (Kuhl, Weatherly & Dahmann, 2000), as are CORBA, Microsoft's .NET, and Enterprise Java Beans.

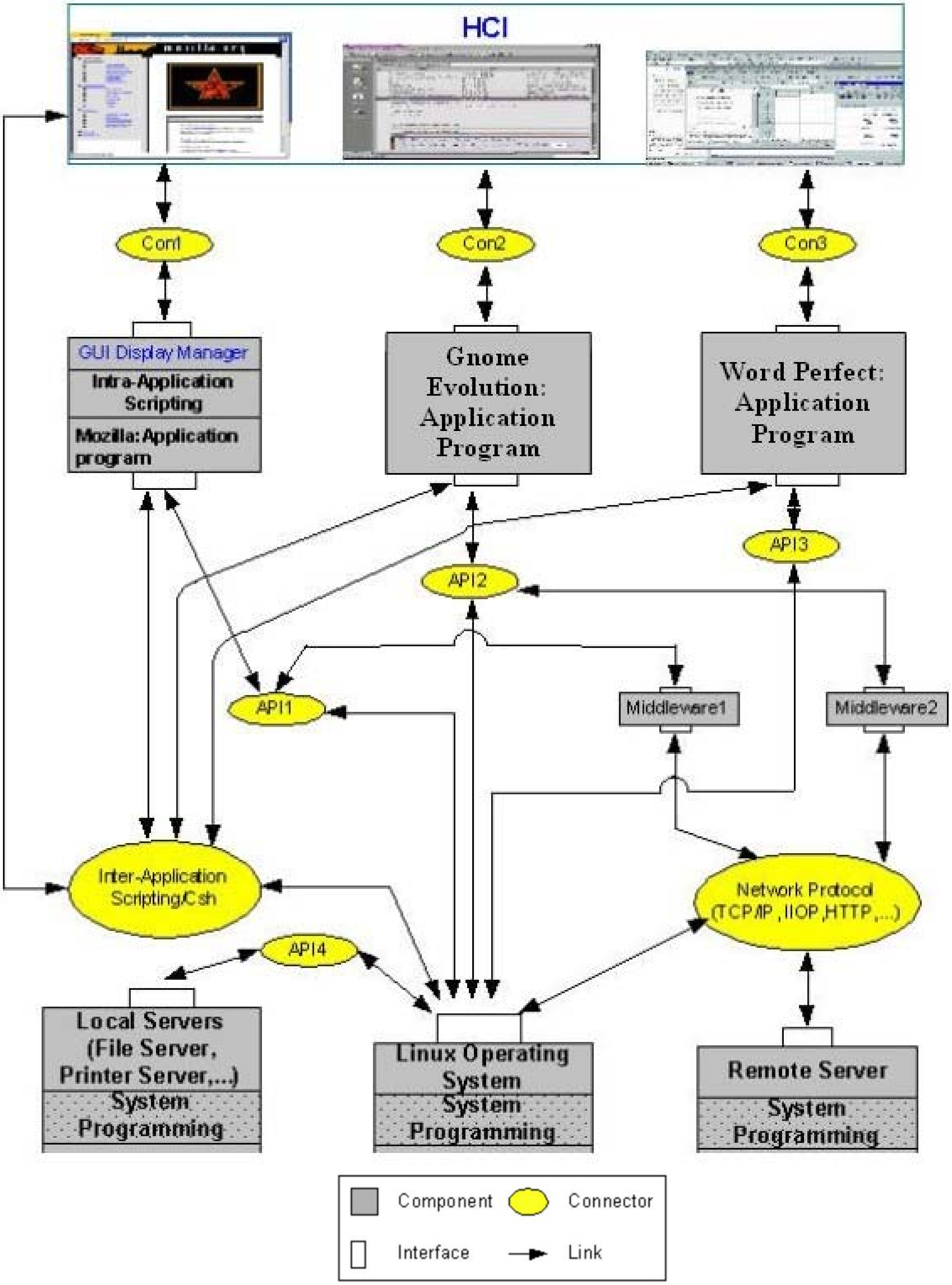
- *Configured system or sub-system*—These are software systems built to conform to an explicit architectural specification. They include software source code/binary components, APIs, and connectors that are organized in a way that may conform to a known “architectural style” such as the Representational State Transfer (Fielding & Taylor, 2002) for Web-based client-server applications or may represent an original or ad hoc architectural pattern (Bass et al., 2003). All the software elements, and how they are arranged and interlinked, can all be specified, analyzed, and documented using an Architecture Description Language (Bass et al., 2003) and ADL-based support tools. Beyond this, any or all of the software elements in a configured system or sub-system may or may not be OSS. In contrast to a derived work, a configured system or sub-system is considered as a “collective work” and as such is subject to its own copyright and license protection as intellectual property, whether open or closed (Rosen, 2005; St. Laurent, 2004). However, such intellectual property declaration cannot employ a license regime on the overall system that supercedes or controverts the license protections/obligations of the individual software elements that constitute the configured system or sub-system.

Figure 1 provides an overall view of a hypothetical software architecture for a configured system that includes and identifies each of the software elements above. It also includes open source (e.g., Gnome Evolution) and closed source software (WordPerfect) components. In simple terms, the configured system consists of software components (grey boxes in the figure) that include a Mozilla Web browser, Gnome Evolution e-mail client, and WordPerfect word processor that run on a Linux operating system that can access file, print, and other remote networked servers (e.g., Apache Web server). These components are interrelated through a set of software connectors (ellipses in the figure) that connect the interfaces of software components (small white boxes attached to a component) that are linked together. Modern enterprise systems or command and control systems will generally have more complex architectures and a more diverse mix of software components than shown in the figure here. As we examine next, this simple architecture raises a number of OSS licensing issues that mitigate the extent of openness that is realized in a configured OA.

Understanding Open Software Licenses

A particularly knotty challenge is the problem of licenses in OSS and OA. There are a number of different OSS licenses, each with different rights and obligations attached to software components that bear it. External sources are available that describe and explain the many different licenses now in use with OSS (OSI, 2008; Rosen, 2005; St. Laurent, 2004). Thus, we will not delve into the details or variations among the many licenses, except to note a few key properties that should be recognized as potentially impacting the openness of a configured software system, and therefore, whether it can realize an OA.

The GNU General Public License (GPL), the most widely used OSS license, implements a strong copyleft, requiring that the software source code be distributed and that any modified versions also be licensed under GPL (Rosen, 2005; St. Laurent, 2004). The GPL, along with some other OSS licenses like the Mozilla Public License (MPL), and others (CPL, OSL (OSI, 2008; Rosen, 2005)), are identified as “reciprocal” licenses that in some way transfer license obligations to derivative software systems. A software system component or connector based on existing OSS inherits the obligations or restrictions of the originating OSS. In contrast, an academic freedom license such as the BSD, MIT, or Apache license permits derivative software works to be incorporated into a proprietary, closed-source product (Rosen, 2005; St. Laurent, 2004). Academic licenses are identified as “unrestrictive” so that software components or connectors derived from OSS covered by an academic freedom license need not adhere to the obligations of the originating OSS.


HCI
Con1
Con2
Con3
GUI Display Manager
Intra-Application Scripting
Mozilla: Application program
Gnome Evolution: Application Program
Word Perfect: Application Program
API1
API2
API3
API4
Middleware1
Middleware2
Inter-Application Scripting/Csh
Network Protocol (TCP/IP, IIOP, HTTP, ...)
Local Servers (File Server, Printer Server,...)
System Programming
Linux Operating System
System Programming
Remote Server
System Programming
Component
Connector
Interface
Link

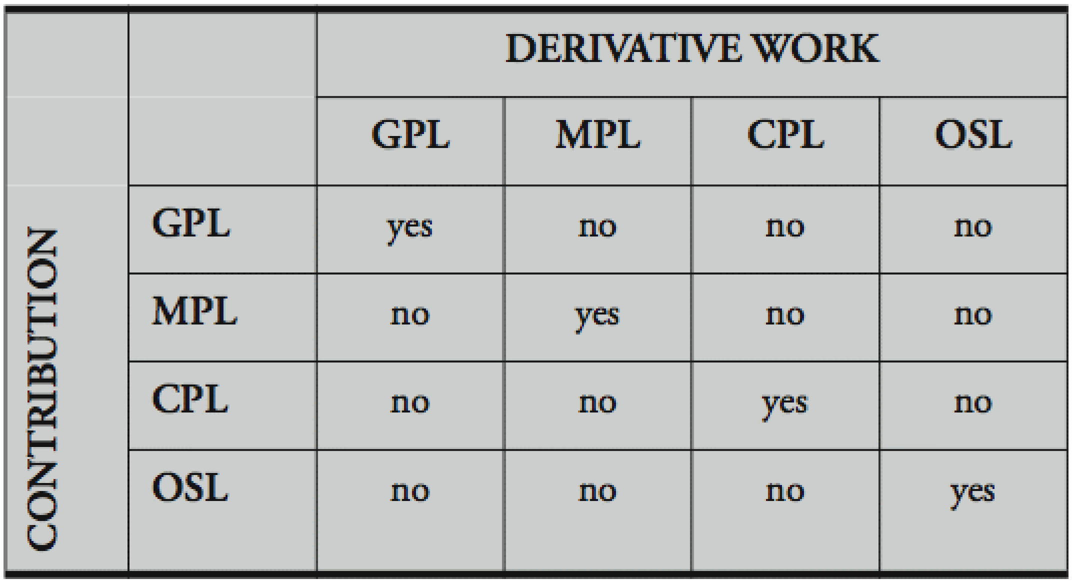
Note: Components, connectors, and overall system configuration may be subject to different software licenses.

Figure 1. Software Components, Connectors, Interfaces Arranged in an Overall Software System Configuration

What license applies to an OA system containing some GPL components with a reciprocal license and some BSD components with unrestrictive license, or perhaps even some proprietary software license? In Figure 1, we see at least three software components that have different software licenses: the Mozilla Web browser (subject to the MPL), Gnome Evolution e-mail client (subject to the GPL), and WordPerfect word processor (subject to a proprietary software license). The license problem is further complicated by components designed to operate on license requirements. For example, a software shim may be a library function, abstract interface, or script code designed to serve as a connector between two applications that have different licenses, so that neither application's license is violated, and neither application is "infected" by the restrictions or obligations of the other's license. In this regard, a software connector is a configured system (or OA) element specifically designed to modulate the license requirements imposed on the components it connects. Figure 1 follows the links between the Mozilla Web browser, Gnome Evolution, and WordPerfect. The requirements imposed by a component's license are affected by the architectural structure of the system containing it and vice versa. Figures 2a and 2b provide suggested mappings of license obligations that can constrain a configured software system derived from OSS components and connectors covered by a specific OSS license.

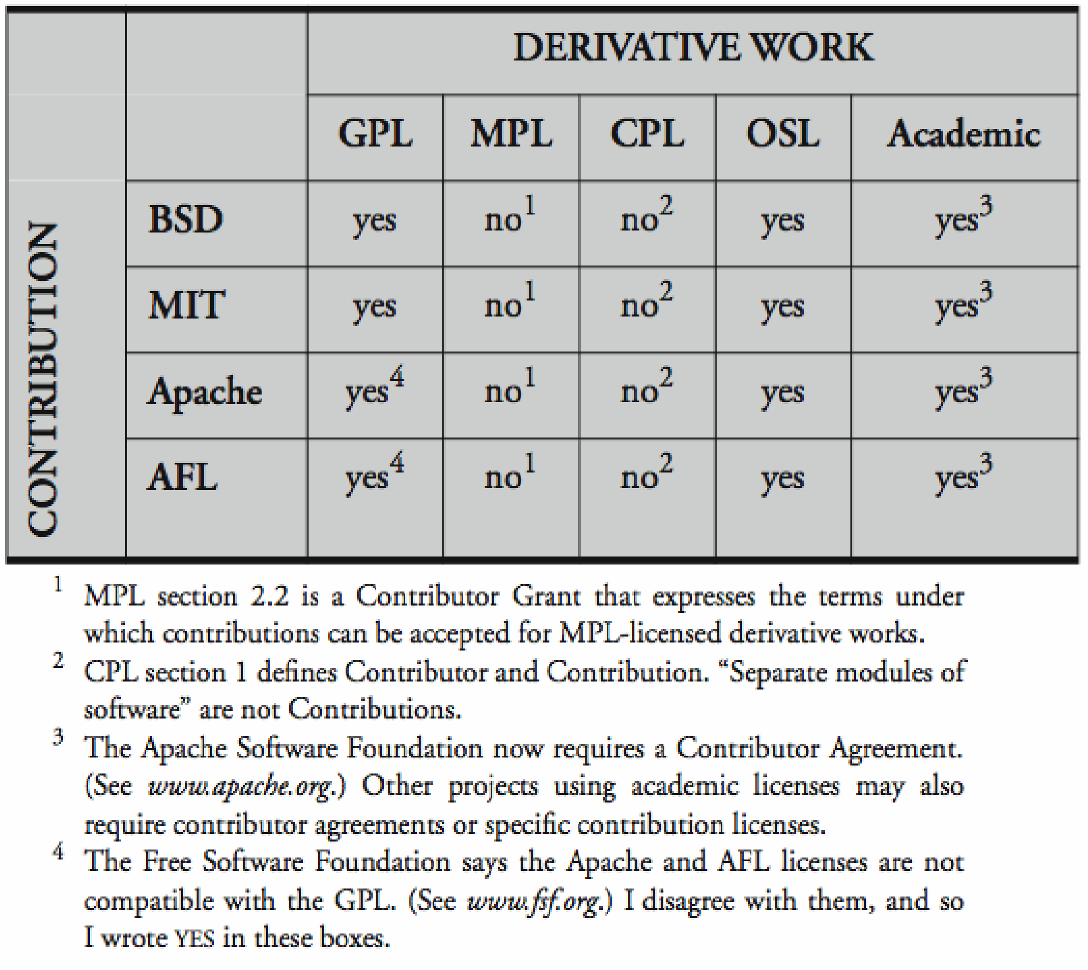
The question of what license covers a specific configured system is difficult to answer, especially if the system or sub-system is already in operation (Kazman & Carrière, 1999). We offer the following considerations to clarify this. For example, a Mozilla/Firefox Web browser covered by the MPL may download and run intra-application script code that is covered by a different license. If this script code is only invoked via dynamic run-time linking (or invocation), then there is no transfer of license restrictions or obligations. However, if the script code is integrated into the source code of the Web browser as persistent part of an application, then it could be viewed as a configured sub-system that may need to be accessed for license transfer implications. Another kind of example can be anticipated with application programs (like Web browsers, e-mail clients, and word processors) that employ Rich Internet Applications or mashups that entail the use of content (e.g., textual character fonts or geographic maps) that is subject to copyright protection—if the content is embedded in and bundled with the scripted application sub-system.

Next, as software system configuration (or OA) is intended to be adapted to incorporate new innovative software technologies that are not yet at hand, we recognize that these OSS-based system configurations will evolve over time at ever-increasing rates (Scacchi, 2007); components will be replaced, and inter-component connections will be rewired or remediated with new connector types. As such, the sustaining the openness of a configured software system will become part of ongoing system support, analysis, and validation. This, in turn, may require ADLs to include OSS licensing properties on components, connectors, and overall system configuration, as well as in appropriate analysis tools (Bass et al., 2003).



		DERIVATIVE WORK			
		GPL	MPL	CPL	OSL
CONTRIBUTION	GPL	yes	no	no	no
	MPL	no	yes	no	no
	CPL	no	no	yes	no
	OSL	no	no	no	yes

Figure 2a. Mapping *Reciprocal* OSS Licenses to Derivative Works
(Rosen, 2005)



		DERIVATIVE WORK				
		GPL	MPL	CPL	OSL	Academic
CONTRIBUTION	BSD	yes	no[1]	no[2]	yes	yes[3]
	MIT	yes	no[1]	no[2]	yes	yes[3]
	Apache	yes[4]	no[1]	no[2]	yes	yes[3]
	AFL	yes[4]	no[1]	no[2]	yes	yes[3]

[1] MPL section 2.2 is a Contributor Grant that expresses the terms under which contributions can be accepted for MPL-licensed derivative works.

[2] CPL section 1 defines Contributor and Contribution. "Separate modules of software" are not Contributions.

[3] The Apache Software Foundation now requires a Contributor Agreement. (See *www.apache.org*.) Other projects using academic licenses may also require contributor agreements or specific contribution licenses.

[4] The Free Software Foundation says the Apache and AFL licenses are not compatible with the GPL. (See *www.fsf.org*.) I disagree with them, and so I wrote YES in these boxes.

Note: Footnotes in original (Rosen, 2005, p. 251).

Figure 2b. Mapping *Unrestrictive* Academic to Reciprocal OSS Licenses

Moving forward, analyses of OSS licenses by intellectual property lawyers may suggest a way out of the current OSS licensing/relicensing mess. Note, we are not lawyers, so we are not offering any legal advice. Feel free to consult legal counsel if or when appropriate for guidance on license interpretation or enforcement conditions. However, we offer some encouraging words. Rosen (2005, p. 252) observes OSS license incompatibilities can prevent OSS from being freely used and combined. The multiplicity of such licenses only makes the problem worse (review the tables in Figure 2a and 2b). Copyright law and contract law which cover the interpretation and enforcement of OSS licenses is such that OSS developers or distributors (e.g., Defense contractors) cannot simply relicense copyright protected OSS unless they have permission to do so. This, in turn, may mitigate some requirements shaping the development and deployment of military software applications that are suppose to embody an OA.

Terms and conditions for reciprocity obligations in licenses like the GPL and others apply to OSS that are modified and redistributed and not to software that may be modified but not distributed outside of the organization. Also, this raises the questions of what constitutes "distribution" or "redistribution" for a government organization that acquires access rights to all software and data developed under contract. Similarly, for government employees whose work is not protected by copyright (and thus may enter into the public domain), this may pose new opportunities for adhering to or working around OSS license restrictions or obligations.

Finally, as Rosen (2005, p. 253) observes, by merely aggregating (or configuring) software from different sources and treating such software as black boxes (e.g., no intra-application scripting allowed and/or employ dynamic run-time linkage), it is possible to technically avoid creation of derivative works that inherit the license restrictions or obligations of the involved software elements. Subsequently, Rosen finds that OSS license incompatibilities are inconveniences rather than barriers, and ultimately, one can get around almost all licensing restrictions by being sufficiently creative and inventive. Thus, there is a need to providing guidance to program acquisition officers, Program Executive Offices, and Defense contractors for how to specify requirements for military software applications that best achieve a cost-effective level of openness, which can enable the maximum possible benefits anticipated. But, without explicit guidance or guidelines, we cannot assume that OA will just happen because of the use of OSS elements and open systems APIs.

With this in mind, we outline some initial guidelines for such requirements.

Discussion

The relationships among open technology, open architecture, open source software requirements, and program acquisition is poorly understood. We can call such a view of OSS: (a) *product oriented*. Alternatively, we can view OSS as: (b) primarily a set of development processes, work practices, project community activities (code sharing, review, modification, redistribution), and multi-project software ecosystem that produce OSS systems and components. This view of OSS as an integrated web of people, processes, and organizations (including project teams operating as virtual organizations (Noll & Scacchi, 1999; Crowston & Scozzi, 2002)) is *production oriented* (including production processes, production organizations, production people, and governance over software production (Scacchi, 2007; Scacchi, Feller, et al., 2006; Scacchi & Jensen, 2008)). The requirements for (a) are not the same as for (b), and program acquisition targeting (a) may fail to realize the benefits, capabilities, or constraints engendered by (b), and vice versa. As such, there is need to understand how to identify an optimal mix of OSS within OA as both products and production processes, practices, community activities, and multi-project (or multi-organization) software ecosystems.

The success of the DoD's OA and OSS programs in achieving the positive qualities associated with OSS depends on the socio-technical context in which a system is developed and used. The stakeholders and users of an OSS system typically include the developers of that system; they know its goals and requirements implicitly and can adapt and evolve the system to follow their understanding of the context in which it is used. If the DoD is to achieve quick response, rapid adaptation, and context-appropriate use of OSS, it may require a representative group of the personnel who use and adapt it to their needs be OSS developers for that system.

Following our analysis above, it appears there are a new set of requirements are emerging that will need to be addressed in any acquisition of a software-intensive system that is stipulated to employ an OA that accommodates OSS components or connectors. PMs that identify specific requirements for a given program acquisition or system development contract can benefit from consideration of the following guidelines for how best to realize an OA:

- Determining how much openness is required or desired.
- Identifying guidelines and incentives for software development contractors that encourage them to develop, provide, and distribute/deploy OA systems with OSS components, connectors, and configuration that minimize conflicting OSS license obligations.
- Determining the restrictions, if any, that apply to the OSS licenses used by different software system components, connectors, or configurations within an OA system.
- Identifying alternative OSS component, connector, or configuration candidates that may satisfy a specified, overall system architecture.
- Determining scenarios that help reveal whether there are OSS licensing conflicts for a given set of OSS components, connectors, or configuration.
- Identifying and analyzing any OSS licensing obligations that must be satisfied for the resulting system to be available for redistribution.
- Identifying and validating OSS license conformance criteria for configured systems intended for redistribution.

Further elaboration on these guidelines is subject to additional research, application, and refinement. However, they do provide a useful starting point for discussion, debate, and action in program acquisition.

Conclusions

The relationships among open technology, open architecture, open source software requirements, and program acquisition is poorly understood. In recent OA presentations, OSS is viewed as primarily a source for low-cost/free software systems or software components. Thus, given the goal of realizing an OA and open technology strategy (Herz & Scott, 2007), together with the use of OSS components and open APIs, it is unclear how to best align program acquisition, system requirements, software architectures, and OSS license regimes. Subsequently, the central problem we examined in this paper was how to identify principles of software architecture and OSS copyright licenses that facilitate or inhibit the success of an OA strategy when OSS and open APIs are required or otherwise employed.

Consideration of emerging issues in the acquisition of OSS within the US Department of Defense is currently an important problem for acquisition research. The goal of this paper is to help establish a foundation for how to analyze and evaluate dependencies that might arise when one is seeking to develop software systems that should embody an OA and when different types of OSS components or OSS component licenses are being considered for integration.

List of References

Alspaugh, T.A., & Antón, A.I., (2007). Scenario support for effective requirements. *Information and Software Technology*, *50*(3), 198-220.

Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice* (2nd ed.), New York: Addison-Wesley Professional.

Bollinger, T. (2003, January 2). *Use of free and open-source software (FOSS) in the US Department of Defense.* The MITRE Corporation. Retrieved March 2008, from http://www.terrybollinger.com/dodfoss/dodfoss_html/index.html

Choi, J.S., & Scacchi, W. (2001, December 15). Modeling and simulating software acquisition process architectures. *Journal of Systems and Software*, *59*(3), 343-354.

Crowston, K., & Scozzi, B. (2002). Open source software projects as virtual organizations. *IEEE Proceedings—Software*, *149*(1), 3-17.

Diallo, M., Sim, S.E., & Alspaugh, T.A. (2007). The mythical requirements-architecture gap. Submitted to the 2007 European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE).

Feldt, K. (2007). *Programming Firefox: Building rich internet applications with XUL.* Sebastopol, CA: O'Reilly Press.

Fielding, R., & Taylor, R.N. (2002). Principled design of the modern web architecture. *ACM Transactions Internet Technology*, *2*(2), 115-150.

Guertin, N. (Director, Open Architecture, Program Executive Office IWS 7B). (2007, March 14). *Naval open architecture: Open architecture and open source in DOD*. Remarks delivered at “Open Source—Open Standards—Open Architecture,” Association for Enterprise Integration Symposium, Arlington, VA.

Herz, J.C., & Scott, J. (2007, June). COTR warriors: Open technologies and the business of war. *The DoD Software Tech News*, *10*(2), 3-6. Retrieved March 2008, from https://www.softwaretechnews.com/stn_view.php?stn_id=42

Justice, N., Brig. Gen., Program Executive Office C3T. (2007, March 14). *Open source software challenge: Delivering warfighter value*. Remarks delivered at “Open Source—Open Standards—Open Architecture,” Association for Enterprise Integration Symposium, Arlington, VA.

Justice, N., Brig. Gen., Program Executive Office C3T. (2007, December 12). *Deploying open technologies and architectures within military systems*. Remarks delivered at 3rd DoD Open Conference, Deployment of Open Technologies and Architectures within Military Systems, Association for Enterprise Integration Symposium, Arlington, VA.

Kazman, R., & Carrière, J. (1999). Playing detective: Reconstructing software architecture from available evidence. *Journal of Automated Software Engineering, 6*(2), 107-138.

Kuhl, F., Weatherly, R., & Dahmann, J. (2000). *Creating computer simulation systems: An introduction to the high level architecture*. Upper Saddle River, NJ: Prentice-Hall PTR.

Meyers, B.C., & Obendorf, P. (2001). *Managing software acquisition: Open systems and COTS products*. New York: Addison-Wesley.
Nelson, L., & Churchill, E.F. (2006, September). Repurposing: Techniques for reuse and integration of interactive services. In *Proceedings of the 2006 IEEE International Conference of Information Reuse and Integration.* Los Alamitos, CA: IEEE.

Noll, J., & Scacchi, W. (1999, February). Supporting software development in virtual enterprises. *Digital Information*, *1*(4).

OSI. (2008). *The open source initiative*. Retrieved March 2008, from http://www.opensource.org/

Riechers, C., Principal Deputy, Asst. Sect. of the Air Force, Acquisition. (2007, March 14). *The role of open technology in improving USAF software acquisition*. Remarks delivered at “Open Source—Open Standards—Open Architecture,” Association for Enterprise Integration Symposium, Arlington, VA.

Rosen, L. (2005). *Open source licensing: Software freedom and intellectual property law*. Upper Saddle River, NJ: Prentice-Hall PTR. Retrieved from http://www.rosenlaw.com/oslbook.htm

Scacchi, W. (2002, February). Understanding the requirements for developing open source software systems, *IEE Proceedings—Software*, *149*(1), 24-39.

Scacchi, W. (2007). Free/Open source software development: Recent research results and methods. In M. Zelkowitz (Ed.), *Advances in Computers*, *69*, 243-295.

Scacchi, W., Feller, J., Fitzgerald, B., Hissam, S., & Lakhani, K. (2006, March/April). Understanding free/open source software development processes. *Software Process—Improvement and Practice*, *11*(2), 95-105.

Scacchi, W., & Jensen, C. (2008). *Governance in open source software development projects: Towards a model for network-centric edge organizations.* Remarks delivered at the 13th International Command and Control Research and Technology Symposium, Bellevue, WA. To appear in June.

Starrett, E. (2007, May). Software acquisition in the army. *Crosstalk: The Journal of Defense Software Engineering*, 4-8. Retrieved March 2008, from http://stsc.hill.af.mil/crosstalk

St. Laurent, A.M. (2004). *Understanding open source and free software licensing*. Sebastopol, CA: O'Reilly Press.

US Navy. (2006). *Naval OA strategy*. Retrieved March 2008, from https://acc.dau.mil/oa

Weathersby, J.M. (2007, June). Open source software and the long road to sustainability within the US DoD IT system. *The DoD Software Tech News*, *10*(2), 20-23. Retrieved March 2008, from https://www.softwaretechnews.com/stn_view.php?stn_id=42

Wheeler, D.A. (2007, June). Open source software (OSS) in US government acquisitions. *The DoD Software Tech News*, *10*(2), 7-13. Retrieved March 2008, from https://www.softwaretechnews.com/stn_view.php?stn_id=42

Acknowledgments

The research described in this report has been supported by grants #0534771 from the US National Science Foundation and the Acquisition Research Program at the Naval Postgraduate School. No endorsement implied.

Panel 10 - Collaborative Product Lifecycle Management & 3-D Visualization Tools: Implications for the SHIPMAIN Process

Wednesday, May 14, 2008 **3:30 p.m. – 5:00 p.m.**	**Panel 10 - Collaborative Product Lifecycle Management & 3-D Visualization Tools: Implications for the SHIPMAIN Process**
	Chair: **Ms. Lorna B. Estep**, Deputy Director for Supply, Directorate of Logistics Sustainment, Headquarters, Air Force Materiel Command **Discussant:** **Mr. Michael Schwind**, Vice President, Maritime Solutions Sector, Siemens, Product Lifecycle Management Software **Papers:** ***The Potential Impact of Collaborative and Three-dimensional Imaging Technology on SHIPMAIN Fleet Modernization Plan*** **LT Nate Seaman, United States Navy**, Head of Information Management, Camp Pendleton ***The Use of Collaborative and Three-dimensional Imaging Technology to Achieve Increased Value and Efficiency in the SHIPMAIN Cost Estimation Process*** **Dr. Johnathan Mun**, Professor of Research, Naval Postgraduate School

Chair: Lorna B. Estep, a member of the Senior Executive Service, is Deputy Director for Supply, Directorate of Logistics, Headquarters Air Force Materiel Command, Wright-Patterson Air Force Base, Ohio. She is responsible for the Materiel Support Division of the Supply Management Activity Group, a stock fund with annual sales of $7 billion. She directs a wide range of logistics services in support of Air Force managed spare parts, to include transformation programs, requirements determination, budgeting, acquisition, provisioning, cataloging, distribution and data management policy. She also provides supply chain management policy, guidance and direction in support of headquarters, air logistics centers, and US Air Force worldwide customers.

Estep started her career as a Navy logistics management intern. She has directed the Joint Center for Flexible Computer Integrated Manufacturing, was the first program manager for Rapid Acquisition of Manufactured Parts, and has served as Technical Director of Information Technology Initiatives at the Naval Supply Systems Command. In these positions, she has developed logistics programs for the Department of Defense, implemented one of the first integrated and agile data-driven manufacturing systems, and directed the development of complex technical data systems for the Navy.

As the Director of Joint Logistics Systems Center, Estep had the duties of a commanding officer for a major subordinate command. In addition, she acted as the Logistics Community Manager, an emerging organization to coordinate and implement the revised Defense Department logistics strategy for achieving Joint Vision 2010 through modern information techniques and processes. She has also served as Chief Information Officer for the Naval Sea Systems Command in Arlington, VA, and Executive Director of Headquarters Materiel Systems Group at Wright-Patterson AFB. Prior to her current assignment, she served as Deputy Director for Logistics Readiness at the Pentagon, where she developed combat support concepts, doctrine, and sustainment policy with the Office of the Secretary of Defense, defense agencies, the Joint Chiefs of Staff and combatant commanders.

2003 - 2008 Sponsored Research Topics

Acquisition Management

- Software Requirements for OA
- Managing Services Supply Chain
- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Portfolio Optimization via KVA + RO
- MOSA Contracting Implications
- Strategy for Defense Acquisition Research
- Spiral Development
- BCA: Contractor vs. Organic Growth

Contract Management

- USAF IT Commodity Council
- Contractors in 21st Century Combat Zone
- Joint Contingency Contracting
- Navy Contract Writing Guide
- Commodity Sourcing Strategies
- Past Performance in Source Selection
- USMC Contingency Contracting
- Transforming DoD Contract Closeout
- Model for Optimizing Contingency Contracting Planning and Execution

Financial Management

- PPPs and Government Financing
- Energy Saving Contracts/DoD Mobile Assets
- Capital Budgeting for DoD
- Financing DoD Budget via PPPs
- ROI of Information Warfare Systems
- Acquisitions via leasing: MPS case
- Special Termination Liability in MDAPs

Human Resources

- Learning Management Systems
- Tuition Assistance
- Retention
- Indefinite Reenlistment

- Individual Augmentation

Logistics Management

- R-TOC Aegis Microwave Power Tubes
- Privatization-NOSL/NAWCI
- Army LOG MOD
- PBL (4)
- Contractors Supporting Military Operations
- RFID (4)
- Strategic Sourcing
- ASDS Product Support Analysis
- Analysis of LAV Depot Maintenance
- Diffusion/Variability on Vendor Performance Evaluation
- Optimizing CIWS Lifecycle Support (LCS)

Program Management

- Building Collaborative Capacity
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to Aegis and SSDS
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Terminating Your Own Program
- Collaborative IT Tools Leveraging Competence

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.org

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.org